TIME MANAGEMENT INCLUDING TWO-MACHINE FLOWSHOP SCHEDULING WITH INTERVAL PROCESSING TIMES

Yuri N. Sotskov[1], Natalja M. Matsveichuk[1], Frank Werner[2], Andrei A. Kasiankou[1]

[1] United Institute of Informatics Problems, Surganov Str. 6, 220012 Minsk, Belarus
emails: sotskov@newman.bas-net.by, matsveichuk@tut.by, aklotr809@gmail.com

[2] Otto-von-Guericke-Universität, Fakultät für Mathematik, PSF 4120, 39016 Magdeburg, Germany
email: frank.werner@ovgu.de

Abstract: *The purpose of this article is to initiate a research in one of the main parts of time management, which is an extremely useful activity that implies an effective use of personal time. Namely, we propose to include optimal scheduling algorithms into time management techniques. Specifically, we consider a time management system including two persons, one of which is a supervisor. In scheduling theory, such a processing system may be modeled as a two-machine flowshop with the objective of minimizing the makespan. Due to the specificity of the scheduling process for a person, it is assumed that the job processing times may be uncertain before scheduling. Moreover, for time management, it is necessary to consider on-line scheduling along with usual off-line scheduling for a set of jobs to be completed by a person during a day.*

Keywords: *Scheduling, flowshop, time management.*

MSC*: 90B35,90B36.*

April 22, 2014

Introduction

Despite of the large number of publications on time management and its great popularity in many countries, the definition of time management varies in different sources. In [Hellsten, 2012], the following definition of time management has been provided: "Time management is the analysis of how working hours are spent and the prioritization of tasks in order to maximize personal efficiency in the workplace. In more specialized sources, and depending on the scope of the notion of the term may be supplemented." An ultimate goal of time management is to increase the efficiency of time usage by a person. Time management deals with a set of rules, methods, and principles that may be used to increase the personal efficiency due to a better time usage.

The main steps for personal time optimization include setting goals, compiling lists of tasks, and personal planning in short and long time periods. The variety of the literature on time management does not mean that there already exists a theoretical framework containing a basic set of statements, formal definitions, rules, and methods such that they are sufficient for all people and that this knowledge allows every person to know how to manage his time effectively.

The modern literature on time management presents mainly empirical rules, the use of which can help some persons to manage their time more efficiently. However, if these rules do not help a concrete person, it is difficult to assert whether this is a result of improper performance of the proposed time management rules and techniques or whether these rules and proposed time management technology are not suitable for the particular person.

After studying techniques of time management and starting to implement them, a person may need a variety of supporting tools. For example, following the time management recommendations and principles, a person may need to make a list of tasks that expose their priorities, sort them according to their urgency, possible profit, time needed, etc. Of course, one can use a pen and a sheet of paper to do this. Indeed in many cases it will be justified if suddenly urgent tasks appear, then one can write the tasks on the sheet of paper and then execute the tasks. However, if the most amount of information that a person uses when planning the working time is quite large, it is necessary to use more sophisticated tools, e.g., electronic devices and their software.

Today, there are a wide variety of computer applications, which are more or less designed to help a person to manage time to execute his to-do-list. Unfortunately, to the best of our knowledge, all these computer systems cannot construct an optimal schedule for executing the to-do-list. If a person is a leader of a team and works with a few other people, then it is needed to use a computer application such that time optimization takes into account the subordination of the team members.

Literature review

The literature dealing with time management can be divided into two main parts: the popular literature and scientific articles. Books on time management, brochures, popular science articles and materials of numerous seminars on time management are included into the popular literature, which provide a variety of information about time management, such as recommendations, expert advices, analysis, and applications of advertising time management. Despite of the large number of publications on time management and its popularity [Jackson, 2009] – [Macan, 2010], the definitions, terminology and content of time management in a variety of references vary widely. With certainty, it can be argued that the basic principles of time management are as follows: the need for setting the general and immediate goals and objectives for a person, planning (long-term and short-term) work, and compiling lists of works to be performed at a certain time.

The existence of a large number of references does not mean that currently a deep theoretical basis of time management is already developed, which would contain a complete set of models, provisions, definitions and methods, which are sufficient so that a top executive could successfully use existing tools of time management planning for the working time. The literature on time management provides descriptions of empirical rules, which are intended to help a person to manage their working time. If the efficiency of the worker does not increase as a result of the implementation of the recommended tools and rules of time management, it is usually impossible to ascertain whether this is a result of improper performance of the proposed rules and techniques of time management, or whether it is a result of the fact that this method of time management for a particular individual is generally ineffective.

A number of articles evaluate the effectiveness of a particular method of time management. It should be noted that, as a scientific discipline in the articles on time management, psychology is often used, so in most scientific papers, time management is considered in the context of individual empowerment and human psychophysical characteristics. For example, in [Jackson, 2009], a methodology and principles of time management are provided, the use of which is intended to increase the effectiveness of human activity.

In [Woolfolk, 1986], an experiment conducted in a college is described in detail, in which the effect of time management on the productivity of students is assessed. In the experiment, the students were divided into groups, one of which offered a course on time management. In another group of students, together with a course on time management, workshops on time management were conducted (compiling lists of tasks, planning their implementation). In the control group of students, lessons on time management were conducted. The experimental results showed convincingly that in addition to improving students' knowledge, sessions on time management had a positive impact on the performance of the students in their training at the college and the completion of all training plans and assignments.

In [Ho, 2003], the results of studies were considered, which showed that four students in the learning process differently interpreted and applied the same time management recommendations. In [Kelly, 2003], the question of how to use time management can influence the level of anxiety and concern of a person was investigated. The studies found that structuring and streamlining the working time as well as the timely setting goals and objectives lead to a decrease in the level of nervousness and anxiety of a person.

In [Macan, 2010], the impact of procedures for the preparation of plans, setting goals and priorities for the development of objectives of human memory was examined. In [Keniga, 2013], the efficacy of "quiet time" (the time of the supervisor that is free of calls, visitors, letters, etc.) is assessed, and it has been experimentally found that the use of "quiet hours" leads to an improved performance of the leader.

In [Zampetakis, 2010], the effect of time management on the creative activity of a person was examined. The study showed that the application of the principles of time management has a positive impact on the creative process. An application of time management to the management of various public events was considered in [Ahmad, 2012]. The study showed that there is a close relationship between the efficiency of social events and the use of technology for their time management. Time management is especially useful in the cases of deadlines and public events when the event requires to perform a large amount of work. Many scientific articles describing experiments were focused on the study of the influence of an individual empowerment and human psychophysical characteristics on the development and effective use of time management techniques.

In scheduling theory, models and algorithms have been developed to construct optimal (or approximate) schedules for a deterministic or stochastic system consisting of a single machine or multiple identical or different machines. For the use of existing tools in the planning of employment rights, deterministic algorithms for solving problems that are investigated in the classical scheduling theory are insufficient. The specificity of an optimal planning of the working time is that in practice, the governing employee has to solve previously unscheduled tasks. A person needs to take into account changes of the situation in a timely manner to overcome problems that require a rapid adjustment of the current schedule of the tasks related to changes as a set of planned works, and sometimes with the emergence of new non-standard problems and issues.

Practical problems of time management are characterized by an uncertain duration of the planned tasks and works because in a schedule for a person, it is often not possible to accurately determine the duration of all jobs (tasks, works) to be scheduled. However, it is generally possible to determine accurate lower and upper bounds of the jobs, i.e., it can be assumed that intervals of possible values of the planned duration of the procedures are known in advance. In this regard, for time management acceptable methods for constructing schedules for service systems with uncertain (interval) numerical parameters are important. Feature systems serving time management should be considered at various stages of the preparation and implementation of schedules, either directly for the statement of the problems of working time planning or in scheduling algorithms for a working day or longer periods of time.

The allocated time management problem can be solved at a large extent by using appropriate models and methods for solving (in some sense) uncertain scheduling problems that have been considered in the last decade [Sotskov, 2010; Sotskov, 2014]. The classical deterministic formulation and the solution methods for the corresponding scheduling problems have been modified by taking into account the uncertainty of the length of an operation which turned out to be important for solving practical time management. Studies of service systems such as "master – subordinate employer" ones are necessary and of particular interest when an interval for the duration of the works is given. In this case, one needs to solve the problem of scheduling the work performed for the serving system, in which a chief may delegate certain tasks to a subordinative person, and the flow of work is directed in one direction: from the main worker to directly subordinate employees.

In the next section, we formulate the formal problem setting and present possible approaches to its exact or approximate solution.

Flowshop scheduling with two machines and interval job processing times

The scheduling problem for time management can be formulated in the following way. There are two employees, a supervisor (or master) and a subordinate person, and a set of jobs to be processed (during a working day, or during a week, etc.). First, every job has to be processed by the master (the first operation on the job). For example, the master poses the task, outlines its solution, defines the stages of the job, and then sells the job to a subordinate person. Then a subordinate employee has to process the job given to him by his supervisor (the second operation on the job) and after that the whole job is completed.

An important feature for a time management schedule is that an exact processing time of the human job may be unknown. Usually, only a lower bound and an upper bound for the processing time of each job are known before scheduling. In time management, we are dealing with a real person and not with a machine. It is always possible to set a reasonable limit for the processing time of each job, at the worst, one can set the lower bound to be equal 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). Another important point is the scheduling goal: we can assume that all necessary jobs should be completed as soon as possible.

Now we can formulate the problem in terms of scheduling theory as follows. Let two machines $\mathcal{M} = \{M_1, M_2\}$ be given to process $n \geq 2$ jobs $\mathcal{J} = \{J_1, J_2, ..., J_n\}$, which have to follow the same machine route. Each job $J_i \in \mathcal{J}$ has to be processed first by machine M_1 and then by machine M_2 without preemption on each machine. All the n jobs are available to be processed from time 0. The factual processing time of job $J_i \in \mathcal{J}$ on machine $M_j \in \mathcal{M}$ is denoted as p_{ij}. The processing time p_{ij} is unknown before scheduling. In the realization of the process, p_{ij} may take any real value from the (closed) interval $[p_{ij}^L,\ p_{ij}^U]$, $p_{ij}^L < p_{ij}^U$, where the lower bound p_{ij}^L and the upper bound p_{ij}^U are fixed before scheduling, but the probability distributions of the random processing times between these lower and upper bounds are unknown. Let $C_i(\pi)$ denote the completion time of job $J_i \in \mathcal{J}$ in the schedule π, and the criterion C_{max} denotes the minimization of the schedule length $C_{max}(\pi)$:

$$C_{max} = \min_{\pi \in \Omega} C_{max}(\pi) = \min_{\pi \in \Omega} \{\max\{C_i(\pi) \mid J_i \in \mathcal{J}\}\},$$

where Ω denotes the set of semi-active schedules [Sotskov, 2014] with the cardinality $|\Omega| = (n!)^2$. In a semi-active schedule, the processing of each job $J_i \in \mathcal{J}$ starts as early as possible (provided that the order of the jobs $\mathcal{J}$ for processing is determined in the schedule). Such a two-machine minimum-length flowshop scheduling problem with *interval* processing times is called an *uncertain* problem and is denoted by $F2|p_{ij}^L \le p_{ij} \le p_{ij}^U|C_{max}$. Let

$$T = \{p \mid p_{ij}^L \le p_{ij} \le p_{ij}^U,\ J_i \in \mathcal{J},\ M_j \in \mathcal{M}\}$$

be the set of all possible vectors (scenarios) $p = (p_{1,1}, p_{1,2}, \dots, p_{n1}, p_{n2})$ of the job processing times. For a fixed vector $p \in T$, the uncertain problem $F2|p_{ij}^L \le p_{ij} \le p_{ij}^U|C_{max}$ turns into the classical deterministic problem $F2||C_{max}$ associated with the vector p of job processing times. The problem $F2|p_{ij}^L \le p_{ij} \le p_{ij}^U|C_{max}$ corresponds to the above time management scheduling problem. The machine M_1 corresponds to the master employee while the machine M_2 corresponds to the subordinate.

Concerning interruptions, we assume that, by processing any job, the employee processes it up to the end without an interruption. Of course, in real life, the employees can be interrupted during the processing of jobs, but this can only lead to an increase of the processing time (the employee will need to "tune" this job again after each interruption). Moreover, interruptions cannot reduce the C_{max} value for the problem $F2||C_{max}$ [Sotskov, 2014]. The above processing times p_{ij} are uncertain.

In this paper, we assume that any processing time cannot be determined accurately (there always exists at least a computational error). So, in time management, the inequalities $p_{ij}^L < p_{ij}^U$, $J_i \in \mathcal{J}$, $M_j \in \mathcal{M}$, hold. Therefore, for finding an optimal order of the job execution by the master employee and the subordinate, we can use the results obtained for the uncertain flowshop problem $F2|p_{ij}^L \le p_{ij} \le p_{ij}^U|C_{max}$. This problem has been investigated in [Matsveichuk, 2014; Matsveichuk, 2009; Matsveichuk, 2011] for the case of $p_{ij}^L \le p_{ij}^U$.

Next, we present some results, which are modified for the case under consideration with $p_{ij}^L < p_{ij}^U$. Let $S = \{\pi_1, \pi_2, \dots, \pi_{n!}\}$ be the set of all permutations of the n jobs from the set $\mathcal{J}$:

$$\pi_k = (J_{k_1}, J_{k_2}, \dots, J_{k_n}), \{k_1, k_2, \dots, k_n\} = \{1, 2, \dots, n\}.$$

The set S defines all *permutation* schedules that *dominate* the set Ω of semi-active schedules for the problem $F2||C_{max}$ associated with each vector p. Thus, the set S of permutation schedules is *dominant* for the uncertain problem $F2|p_{ij}^L \le p_{ij} \le p_{ij}^U|C_{max}$. There exists at least one optimal semi-active schedule with the same sequence of the jobs on both machines M_1 and M_2 [Sotskov, 2014] (i.e., the master employee and the subordinate process the jobs in the same order). Therefore, it is sufficient to look for an optimal schedule among the set S of permutation schedules [Sotskov, 2014]. Since each permutation $\pi_k \in S$ uniquely defines the set of the *earliest* completion times $C_i(\pi_k)$ of the jobs $J_i \in \mathcal{J}$ for the problem $F2||C_{max}$, we identify a *permutation* $\pi_k \in S$ with a *permutation schedule* defined by π_k. The set of all permutation schedules has the cardinality $|S| = n!$. Next, we restrict further the set of permutations that are sufficient to be examined while solving the problem $F2|p_{ij}^L \le p_{ij} \le p_{ij}^U|C_{max}$.

In [Johnson, 1954], it was proven that a permutation $\pi_i = (J_{i_1}, J_{i_2}, \dots, J_{i_n}) \in S$ with

$$\min\{p_{i_k 1}, p_{i_m 2}\} \le \min\{p_{i_m 1}, p_{i_k 2}\}, \tag{1}$$

$1 \le k < m \le n$, is optimal for the deterministic problem $F2||C_{max}$. Each permutation π_i satisfying condition (1) is called a Johnson permutation. We consider a *minimal dominant set* of permutations instead of the whole set S.

Definition 1. *The set of permutations $S(T) \subseteq S$ is called a J-solution (or a minimal dominant set) to the uncertain problem $F2|p_{ij}^L \le p_{ij} \le p_{ij}^U|C_{max}$, if for each vector $p \in T$, the set $S(T)$ contains at least one permutation that is a Johnson one for the deterministic problem $F2|p|C_{max}$ associated with the vector p of job processing times, provided that any proper subset of set $S(T)$ is not a J-solution to problem $F2|p_{ij}^L \le p_{ij} \le p_{ij}^U|C_{max}$.*

Off-line phase of scheduling (constructing a minimal dominant set)

From Definition 1, it follows that a minimal dominant set $S(T)$ contains at least one optimal schedule $\pi_k \in S(T)$ for each vector $p \in T$ of the job processing times, and the set $S(T)$ is a minimal set (with respect to inclusion) which possesses such a property. Thus, to solve problem $F2|p_{ij}^L \le p_{ij} \le p_{ij}^U|C_{max}$ exactly, one can restrict the search within the set $S(T)$ which has often a smaller cardinality than the set S.

The next criterion characterizes the case when $|S(T)| = 1$, i.e., there is an optimal permutation $\{\pi_k \in S\}$ for the problem $F2|p_{ij}^L \le p_{ij} \le p_{ij}^U|C_{max}$ (for the problem $F2||C_{max}$, associated with any vector $p \in T$). Let us consider the following partition of the set $\mathcal{J} = \mathcal{J}_1 \cup \mathcal{J}_2 \cup \mathcal{J}^*$, where

$$\mathcal{J}_1 = \{J_i \in \mathcal{J} \mid p_{i1}^U \le p_{i2}^L, p_{i2}^U > p_{i1}^L\},$$

$$\mathcal{J}_2 = \{J_i \in \mathcal{J} \mid p_{i1}^U > p_{i2}^L, p_{i2}^U \le p_{i1}^L\},$$

$$\mathcal{J}^* = \{J_i \in \mathcal{J} \mid p_{i1}^U > p_{i2}^L, p_{i2}^U > p_{i1}^L\}.$$

Theorem 1. *[Matsveichuk, 2014] There exists a permutation $\pi_k \in S$ which is optimal for problem $F2|p_{ij}^L \le p_{ij} \le p_{ij}^U|C_{max}$ if and only if*
a) for any pair of jobs J_i and J_j from the set $\mathcal{J}_1$ (from the set $\mathcal{J}_2$, respectively) either $p_{i1}^U \le p_{j1}^L$ or $p_{j1}^U \le p_{i1}^L$ (either $p_{i2}^U \le p_{j2}^L$ or $p_{j2}^U \le p_{i2}^L$),
b) $|\mathcal{J}^*| \le 1$ and for job $J_{i^*} \in \mathcal{J}^*$ (if any), the following inequalities hold:

$$p_{i^*1}^L \ge \max\{p_{i1}^U \mid J_i \in \mathcal{J}_1\},\ p_{i^*2}^L \ge \max\{p_{j2}^U \mid J_j \in \mathcal{J}_2\}.$$

Theorem 1 characterizes the case when there exists an optimal order of the jobs in spite of the unknown actual processing times. In such a case, a minimal dominant set $S(T)$ is a singleton: $|S(T)| = 1$.

In the general case, for the uncertain problem $F2|p_{ij}^L \le p_{ij} \le p_{ij}^U|C_{max}$, there often does not exist a single permutation of the n jobs $\mathcal{J}$, which remains optimal for all possible realizations of the job processing times. On the other hand, the cardinality of the set $S(T)$ may be large. The following claim has been proven in [Matsveichuk, 2014].

Theorem 2. *[Matsveichuk, 2014] If inequality*

$$\max\{p_{ik}^L \mid J_i \in \mathcal{J}, M_k \in \mathcal{M}\} < \min\{p_{ik}^U \mid J_i \in \mathcal{J}, M_k \in \mathcal{M}\} \quad (2)$$

holds, then $S(T) = S$.

The above Theorem 2 implies the following claim.

Corollary 1. *Let $p^L_{ik} < p^U_{ik}, J_i \in \mathcal{J}, M_k \in \mathcal{M}$. For any permutation $\pi_k \in S$, there exists a vector $p \in T$ such that permutation π_k is the unique optimal permutation for the problem $F2|p|C_{max}$, associated with vector $p \in T$, if inequality (2) holds.*

On the base of Definition 1, one can fix the order $J_v \to J_w$ of the jobs $J_v \in \mathcal{J}$ and $J_w \in \mathcal{J}$ while solving the problem $F2|p^L_{ij} \leq p_{ij} \leq p^U_{ij}|C_{max}$. Fixing the order of two jobs is possible if, for each vector $p \in T$, there exists a Johnson permutation for the deterministic problem $F2|p|C_{max}$ associated with the vector p of the job processing times with the same order of these two jobs.

Theorem 3. *[Matsveichuk, 2014] There exists a J-solution $S(T)$ to the problem $F2|p^L_{ij} \leq p_{ij} \leq p^U_{ij}|C_{max}$ with the fixed order $J_v \to J_w$ of the jobs $J_v \in \mathcal{J}$ and $J_w \in \mathcal{J}$ in all permutations $\pi_k \in S(T)$ if and only if at least one of the following conditions hold:*

$$p^U_{v1} \leq p^L_{v2} \text{ and } p^U_{v1} \leq p^L_{w1}, \tag{3}$$

$$p^U_{w2} \leq p^L_{w1} \text{ and } p^U_{w2} \leq p^L_{v2}. \tag{4}$$

Next, we show that the set $S(T)$ may be represented in a compact form by a dominance digraph with the set $\mathcal{J}$ of vertices and that it takes $O(n^2)$ time to construct such a dominant digraph. Let $\mathcal{J} \times \mathcal{J}$ denote the Cartesian product of the set $\mathcal{J}$. Due to Theorem 3, by testing inequalities (3) and (4) for each pair of jobs $J_v \in \mathcal{J}$ and $J_w \in \mathcal{J}$, one can construct the following binary relation $\mathcal{A}_\prec \subseteq \mathcal{J} \times \mathcal{J}$ on the set $\mathcal{J}$.

Definition 2. *[Matsveichuk, 2014] The inclusion $(J_v, J_w) \in \mathcal{A}_\prec$ with $v \neq w$ holds if and only if there exists a J-solution $S(T)$ to the uncertain problem $F2|p^L_{ij} \leq p_{ij} \leq p^U_{ij}|C_{max}$ such that job $J_v \in \mathcal{J}$ is located before job $J_w \in \mathcal{J}$ (i.e., $J_v \to J_w$) in all permutations $\pi_k \in S(T)$.*

From Definition 2 and inequality $p^L_{ij} < p^U_{ij}$, it follows that the binary relation $\mathcal{A}_\prec$ on the set $\mathcal{J}$ is antireflective, antisymmetric, and transitive, i.e., it is a strict order. The binary relation $\mathcal{A}_\prec$ defines a dominance digraph $(\mathcal{J}, \mathcal{A}_\prec)$ with the vertex set $\mathcal{J}$ and the arc set $\mathcal{A}_\prec$. The relation $(J_v, J_w) \in \mathcal{A}_\prec$ will be represented as $J_v \prec J_w$. It takes $O(n^2)$ time to construct the digraph $(\mathcal{J}, \mathcal{A}_\prec)$ by testing inequalities (3) and (4) for each pair of jobs from the set $\mathcal{J}$.

Theorem 4. *[Matsveichuk, 2014] Let $\mathcal{J} = \mathcal{J}^* \cup \mathcal{J}_1 \cup \mathcal{J}_2$. There exists at most one component with a cardinality greater than one in the dominance digraph $\mathcal{G}$.*

Theorem 4 means that all the components of the dominance digraph $\mathcal{G}$ (except at most one) are isolated vertices. A permutation $\pi_k = (J_{k_1}, J_{k_2}, \ldots, J_{k_n}) \in S$ may be considered as a total order of the jobs $\mathcal{J}$. A total order defined by the permutation π_k is called a *linear extension* of the partial order $\mathcal{A}_\prec$, if each inclusion $(J_{k_u}, J_{k_v}) \in \mathcal{A}_\prec$ implies the inequality $u < v$. Let $\Pi(\mathcal{G})$ denote the set of permutations $\pi_k \in S$ defining all linear extensions of the partial order $\mathcal{A}_\prec$. In particular, if $\mathcal{G} = (\mathcal{J}, \emptyset)$, then $\Pi(\mathcal{G}) = S$. This case was characterized in Theorem 2.

On the other hand, if

$$|\mathcal{A}_\prec| = \frac{n(n-1)}{2},$$

then $\Pi(\mathcal{G}) = \{\pi_k\}$. A criterion for such a case was given in Theorem 1.

Theorem 5. *[Matsveichuk, 2014] For any vector $p \in T$, the set $\Pi(\mathcal{G})$ contains a Johnson permutation for the problem $F2|p|C_{max}$ associated with the vector p of job processing times.*

Corollary 2. *[Matsveichuk, 2014] There exists a J-solution $S(T)$ to the problem $F2|p^L_{ij} \leq p_{ij} \leq p^U_{ij}|C_{max}$ such that the inclusion $S(T) \subseteq \Pi(\mathcal{G})$ holds.*

The permutation $\pi \in \Pi(\mathcal{G})$ is called *redundant* if there exists a J-solution $S(T)$ for the problem $F2|p_{ij}^L \leq p_{ij} \leq p_{ij}^U|C_{max}$ such that $\pi \in \Pi(\mathcal{G}) \setminus S(T)$. The pair of jobs $J_i \in \mathcal{J}$ and $J_j \in \mathcal{J}$ is called a *conflict pair* of jobs, if neither relation $J_i \preceq J_j$ nor $J_j \preceq J_i$ holds. Due to Definition 2, both conditions $(J_i, J_j) \notin \mathcal{A}_\prec$ and $(J_j, J_i) \notin \mathcal{A}_\prec$ hold for a conflict pair of jobs J_i and J_j. Let the inclusion $J_j \in \mathcal{J}^*$ hold. We denote two sets:

$$\mathcal{J}_j' = \{J_q \in \mathcal{J}_2 \mid \min\{p_{j1}^U, p_{j2}^U\} < p_{q2}^U\} \bigcup \{J_r \in \mathcal{J}_1 \cup \mathcal{J}^* \mid \min\{p_{j1}^U, p_{j2}^U\} \leq p_{r1}^L\}; \quad (5)$$

$$\mathcal{J}_j'' = \{J_w \in \mathcal{J}_1 \mid \min\{p_{j1}^U, p_{j2}^U\} < p_{w1}^U\} \bigcup \{J_u \in \mathcal{J}_2 \cup \mathcal{J}^* \mid \min\{p_{j1}^U, p_{j2}^U\} \leq p_{u2}^L\}. \quad (6)$$

Lemma 1. *[Matsveichuk, 2014] If the inclusions $J_j \in \mathcal{J}^*$, $J_q \in \mathcal{J}_j'$, and $J_w \in \mathcal{J}_j''$ hold, then each permutation of the form $\pi_g = (\ldots, J_q, \ldots, J_j, \ldots, J_w, \ldots) \in \Pi(\mathcal{G})$ is redundant.*

Due to Lemma 1, testing whether permutation $\pi_g \in \Pi(\mathcal{G})$ is a redundant permutation takes $O(n)$ time. Let $\Pi^*(\mathcal{G})$ denote the set of permutations remaining in the set $\Pi(\mathcal{G})$ after deleting all redundant permutations according to Lemma 1.

Theorem 6. *[Matsveichuk, 2014] $\Pi^*(\mathcal{G}) = S(T)$.*

It is clear that testing the condition of Theorem 6 takes $O(n)$ time. Due to Theorem 6, a J-solution can be constructed by deleting all redundant permutations from the set $\Pi(\mathcal{G})$. Since the obtained set $\Pi^*(\mathcal{G})$ is uniquely defined, the following claim is correct.

Corollary 3. *The relation $\mathcal{A}_\prec$ defines a unique J-solution $\Pi^*(\mathcal{G}) = S(T)$ to the problem $F2|p_{ij}^L \leq p_{ij} \leq p_{ij}^U|C_{max}$.*

There are two phases in the scheduling process due to the specificity of time management, namely, the off-line phase (the schedule planning phase) and the on-line phase (the schedule execution phase). The information about the lower and upper bounds for each uncertain processing time is available at the beginning of the off-line phase while the local information on the realization (the actual value) of each uncertain processing time is available once the corresponding operation of a job on a machine is completed. In the off-line phase, based on the knowledge about the bounds of the processing times, one can construct the digraph $\mathcal{G}$ which is a compact form of the strict order $\mathcal{A}_\prec$ and which defines a unique J-solution $S(T)$ of the problem $F2|p_{ij}^L \leq p_{ij} \leq p_{ij}^U|C_{max}$ under consideration.

If $|S(T)| = 1$, then we have one permutation which is optimal for all variations of the job processing times. In this case, the scheduling process stopped at the off-line phase. The employees can process the jobs according to the constructed permutation, and they always obtain an actually optimal schedule (for the actual processing times).

If $|S(T)| > 1$, then for different vectors $p \in T$, different permutations from $S(T)$ may be optimal. In the next section, we describe how to use the strict order $\mathcal{A}_\prec$ and the J-solution $S(T)$ at the on-line phase of scheduling to reach an optimal permutation for the actual vector of the job processing times.

Schedule realization (on-line phase of scheduling)

The result of the off-line phase is a digraph $\mathcal{G} = (\mathcal{J}, \mathcal{A}_\prec)$ that represents a compact representation for a J-solution $S(T)$ of the problem $F2|p_{ij}^L \leq p_{ij} \leq p_{ij}^U|C_{max}$. The aim of the on-line phase is to select a factually optimal permutation of the jobs (solution) from the set $\Pi^*(\mathcal{G}) = S(T)$. Thus, the result of the on-line phase is a linear order on the set $\mathcal{J}$ (one permutation π_i from the J-solution $S(T)$). Therefore, at the on-line phase, a scheduler has to solve a more complicated scheduling problem. A scheduler has to use additional information about the fulfilled jobs. Note, that at the off-line phase, a scheduler has a lot of time for scheduling but only little information. In contrast to the off-line phase, at the on-line phase of scheduling, a scheduler has additional information about the course of the process but little time for scheduling. Note that classical scheduling is usually solved only at the off-line phase.

The set $S(T)$ is only a workpiece of the actual schedule, since only one permutation of the given jobs will be realized. Our approach shows how one can use a set $S(T)$ at the on-line stage to select a schedule, which is an optimal one for the factual processing times. We use the following definition of permutation dominance.

Definition 3. *[Matsveichuk, 2014] The permutation $\pi_u \in S$ dominates the permutation $\pi_k \in S$ with respect to T if the inequality $C_{max}(\pi_u, p) \leq C_{max}(\pi_k, p)$ holds for any vector $p \in T$ of the job processing times. The set of permutations $S' \subseteq S$ is called dominant with respect to T if for each permutation $\pi_k \in S$, there exists a permutation $\pi_u \in S'$ which dominates the permutation π_k with respect to T.*

At the on-line scheduling phase, we consider the following case: *the actual value p^*_{ij} of the job processing time p_{ij} is available at the time-point $t_i = c_j(i)$ when the job J_i is completed by machine M_j.* We note that this case is valid for almost all uncertain scheduling problems. A minimal set of jobs $\mathcal{J}_c \subseteq \mathcal{J}$ is called a *conflict set* of jobs if, for each job J_i from the set $\mathcal{J} \setminus \mathcal{J}_c$, the relation $J_i \prec J_j$ holds for all jobs from the set $\mathcal{J}_c$ (the relation $J_j \prec J_i$ holds for all jobs from the set $\mathcal{J}_c$). We consider the case when an arbitrary number of jobs are conflicting at the same on-line decision-making time-points.

Let the partial strict order $\prec$ over the set $\mathcal{J} = \mathcal{J}^* \cup \mathcal{J}_1 \cup \mathcal{J}_2$ be as follows:

$(J_1 \prec \ldots \prec J_k \prec \{J_{k_1}, J_{k_2}, \ldots, J_{k_r}\} \prec J_{k+1} \prec \ldots \prec J_n)$.

In this case, the set of r jobs $\{J_{k_1}, J_{k_2}, \ldots, J_{k_r}\} \subset \mathcal{J} = \mathcal{J}^* \cup \mathcal{J}_1 \cup \mathcal{J}_2$ is conflicting at the time-point $t_k = c_1(k) \geq 0$. All permutations from the set $S(T)$ look as follows: $(J_1, J_2, ..., J_k, ...)$, i.e., their initial parts are the same until job J_k. Then the master employee can start to execute the schedule and processes the jobs from the set $\{J_1, J_2, ..., J_k\}$. At the time-point t_k, the scheduler (the master employee) has the choice between the conflicting jobs for being processed next (immediately after the job J_k). Let at the time-point t_k the subordinate employee (which corresponds to machine M_2) operate the job J_l.

At the time-point $t_k = c_1(k)$, the set of feasible vectors

$$T(t_k) = \{p \in T \mid p_{i1} = p^*_{i1},\ p_{j2} = p^*_{j2},\ 1 \leq i \leq k,\ 1 \leq j < l\}$$

of the job processing times will be used. We can calculate the lower bound $c_2^L(k)$ for the actual value $c_2(k)$ in the following way (see Fig. 1):

$$c_2^L(k) = c_2(l-1) - c_1(k) + \max\{p^L_{l2},\ c_1(k) - c_2(l-1)\} + \sum_{l+1 \leq j \leq k} p^L_{j2}.$$

We can calculate the following upper bound $c_2^U(k)$:

$$c_2^U(k) = c_2(l-1) - c_1(k) + \sum_{l \leq j \leq k} p^U_{j2}.$$

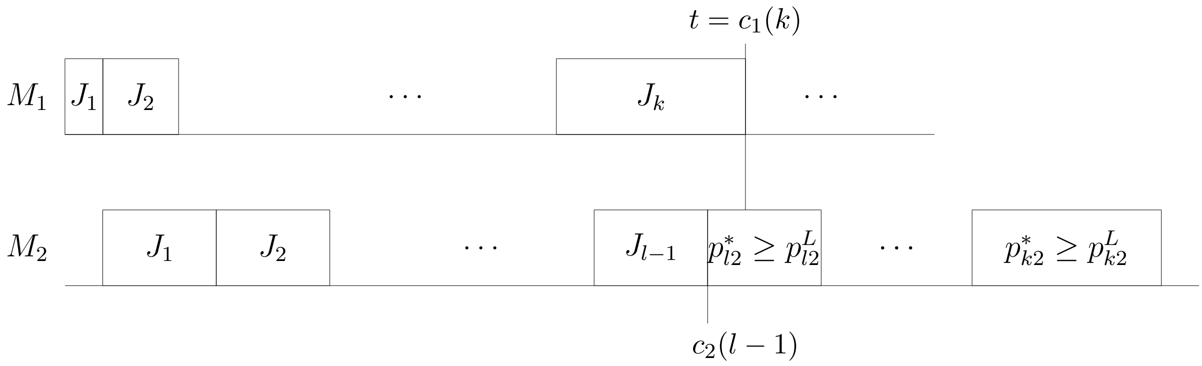

Figure 1: The initial part of an optimal schedule for the jobs from the set $\{J_1, J_2, \dots, J_k\}$

Lemma 2. *[Matsveichuk, 2009] Let the partial strict order $\prec$ over the set $\mathcal{J} = \mathcal{J}^* \cup \mathcal{J}_1 \cup \mathcal{J}_2$ be as follows: $(J_1 \prec \dots \prec J_k \prec \{J_{k_1}, J_{k_2}, \dots, J_{k_r}\} \prec J_{k+1} \prec \dots \prec J_n)$. If the inequality*

$$\sum_{i=1}^{s+1} p^L_{k_i1} \leq \sum_{j=0}^{s} p^U_{k_j2}$$

holds for each $s = 0, 1, \dots, r$, where $p^L_{k_02} = c^L_2(k) - c_1(k)$, then the permutation $\{J_1, \dots, J_k, J_{k_1}, J_{k_2}, \dots, J_{k_r}, J_{k+1}, \dots, J_n\}$ is dominant with respect to $T(t_k)$.

Lemma 3. *[Matsveichuk, 2009] Let the partial strict order $\prec$ over the set $\mathcal{J} = \mathcal{J}^* \cup \mathcal{J}_1 \cup \mathcal{J}_2$ be as follows: $(J_1 \prec \dots \prec J_k \prec \{J_{k_1}, J_{k_2}, \dots, J_{k_r}\} \prec J_{k+1} \prec \dots \prec J_n)$. If the conditions*

$$\sum_{i=m}^{s} p^L_{k_i1} > \sum_{j=m-1}^{s-1} p^U_{k_j2}, \; m = 1, 2, \dots, s,$$

$$\sum_{i=s+1}^{m} p^U_{k_i1} \leq \sum_{j=s}^{m-1} p^L_{k_j2}, \; m = s+1, s+2, \dots, r, \qquad \sum_{i=s+1}^{r+1} p^L_{k_i1} \geq \sum_{j=s}^{r} p^U_{k_j2}$$

hold, where $p^U_{k_02} = c^U_2(k) - c_1(k)$, then the permutation $\{J_1, \dots, J_k, J_{k_1}, J_{k_2}, \dots, J_{k_r}, J_{k+1}, \dots, J_n\}$ is dominant with respect to $T(t_k)$.

Thus, we propose to solve the problem $F2|p^L_{ij} \leq p_{ij} \leq p^U_{ij}|C_{max}$ in two stages as follows. The first step (off-line stage) is the construction of the digraph $\mathcal{G}$ and the check of the conditions of Theorem 1. If there is a permutation, which is a J-solution of the problem $F2|p^L_{ij} \leq p_{ij} \leq p^U_{ij}|C_{max}$, then the solution of the problem is finished at this stage. In the opposite case, we have to go to the second step (on-line stage). At this stage, the employee can start with processing some jobs which are ordered until the first decision point. Then the employee checks the conditions of Lemmas 2, 3, and orders the conflict jobs by using the actual processing times of the already processed jobs. In our approach, to solve the problem $F2|p^L_{ij} \leq p_{ij} \leq p^U_{ij}|C_{max}$ at the decision point, a scheduler indicates the master employee what job to process next. After that, the execution of jobs may be realized until the next decision point, which uses the new additional information about the actual processing times of the jobs, processed between these adjacent decision points.

Time management software

A computer application to manage a list of tasks (jobs) is designed to optimize the working time of the manager by a more rational use. As a method by which this can be achieved, the list of the proposed operations are arranged and evaluated by certain criteria, so that the final list is optimal according to a chosen criterion (e.g., C_{max}). A functional application will provide the possibility to introduce the input (initial) data, to construct an optimal schedule, to monitor the schedule, and to adjust the schedule.

As a schedule for a person implies the need to know what job it is better to do next at the current moment and since a personal schedule is often dynamically changed at a time, the most important features of the graphical user interface should be informative and interactive. A variety of a schedule needs a graphical representation of the schedules under interactivity such that a flexible feedback must be available to inform the user about certain schedule changes and to provide an option of doing the required action and realize a job. Finally, one should pay attention to integrate the principles and approaches of time management into the computer application to maximize the use of personal time. By an integration of time management, it should be understood to systematize the most common "technique" in a compact and user-friendly application. The design and implementation of a personal schedule should be such that the user does not need to think a lot about the technical details of time management and at the same time, an application should allow the user a more efficient use of his time.

Basic information about the architecture of the application and the technologies used

The computer application to manage the tasks of a person based on the model "supervisor – subordinate employer" is a distributed application of a "client-server" (see previous section). This is due to the need to implement the interaction mechanism of a large number of multiple users on a single system, i.e., the behavior, where each user should have access to the shared data, to be able to modify them to a necessary extent, and to be able to inform the other users (other members of the team). With some modifications, the architecture used in our application as the client and on the server side is as follows.

Next, we enumerate each of the layers.

- The layer of the data can be represented by a database that will be used for storing and processing the data, or third-party services, such as data sources. In our case, we use the server database on the server, and a local database as the client. The server database is used for the treatment of the general application data, while the local database is used to store the temporary data on the client, and when the connection to the server is not possible.
- The data access layer is represented by one or more executable libraries and serves to interact with the database, namely, to access, update, and delete data.
- The layer of the domain is represented by one or more executable libraries that are used to describe the application model, the interaction with the data access layer, the implementation of key business scenarios, to implement the communication with various algorithms for constructing schedules, and a proper implementation of the algorithms for constructing schedules, monitoring schedules, adjusting schedules, and statistical calculations.
- The layer of services is represented by one or more executable and libraries used to implement the functional interfaces provided by the server, through which third-party systems can access the functionality implemented by the server.

- The auxiliary layer (proxies), which is represented by one or more executable libraries, serves to integrate with the services of the server. The layer logic of the representations represented by one or more executable library implements the logic representations. In this layer, there will be also the implementation of the data synchronization mechanism for the client and server.
- The presentation layer, which is an implementation of a graphical user interface. There are independent libraries that provide a so-called "cross-cutting ", i.e., it is involved at every level of the application. This includes such features as security authentication, authorization of users, error handling, and system logging application events.

The application is developed based on the platform NET Framework 4.5 in the language of C# 5.0 with 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 development service (in this case, web services) Web server for deploying and running WCF services; WPF - a graphical subsystem for rendering user interfaces in Windows-based applications.

Description of the functionality of the application

We consider the functional application to build schedules for the client and server side. The main feature is that the backbend does not have a GUI, but it implements all the basic logic, to which the end user will receive access from the client. The basic logic of the server presented an algorithm for constructing schedules, monitoring schedules, as well as the functional user authentication, the user registration and the user management.

The client application to build schedules for a person in the hierarchy "leader - subordinate" will provide a different functionality depending on the user's role in the application, namely to be the supervisor or the subordinate. This is due to the fact that for different user roles, the application, except for the general functionality, there will be available a functionality which is specific for each role. The different application functionality will be contained in separate modules that can be connected to the application.

The common functionality for all users is as follows: authentication and authorization; user registration; construction of a schedule; settings; notification service. The common scenario for all users when one first starts the application process is the user registration process and the user authorization. These operations are performed on the server via web services. If the user is already registered in the system, every time one starts the application, it must be authorized.

After the user registration and authorization in the application, the functional configuration becomes available, by means of which the user can specify individual settings: personal settings (personal data), application settings, configure time management, which will then be taken into account in the construction of a schedule. All the settings except for the application settings are stored on the server.

Functional applications for the supervisor

The functional manager has the full functionality for the application management: the user management; to enter the information about the tasks on the basis of which it will be built and the schedule that will be used during the work; constructing schedules; the manual and automatic assignment of tasks performed by users of the schedule; monitoring the implementation of a schedule.

The user management is carried out in a separate module. Here the head is able to manage both a user that has been created, and to create new ones. Under the control of already established means, the users can view their public setting, time management settings, view tasks that are performed by the user, password recovery, as well as to remove a user from the application.

In the future, statistics associated with each user can be realized. The task management module provides a standard mechanism for "create, edit, delete." The GUI data entry forms used are input and output - tables. The schedule management module includes the functional construction of schedules, to assign tasks to users, and to monitor the implementation of a schedule. This is the main application module.

The process of constructing a schedule consists of several steps.

- Receiving raw data from the database; receiving the settings of the subjects involved in the implementation of a schedule (head, subordinates).
- Loading libraries containing the implementation of algorithms that perform a schedule directly. These libraries can be developed dynamically and added to the system without the need for the development of algorithms to be integrated into an application. Algorithms for constructing schedules can be changed in the settings 'Application';
- Constructing a schedule and the automatic assignment of the right of workers to execute appropriate tasks, they can send a notification.

Functional applications for the subject

The user functional at this stage of the development is a minimal set of operations required to perform the task schedules, namely: view tasks necessary to run, including all information that is necessary for their implementation; the opportunity to arrange changes in the status of tasks. After completing the process of the head of a schedule, the slave receives a set of tasks assigned to it and which should be processed. These tasks appear with him in the task list with the status "Not Started."

At the beginning of the task, the employee must change the status of the work to "In Progress." If for some reason the employee does not change the status of the implementation, the system will do this automatically based on the time parameter for the task. At the completion of the work, the subordinates can mention the work done and how to proceed to the next step, or specify the reasons why it cannot be done, and put its execution. In both cases, executives receive a proper notification.

Conclusion

It should be noted that the use of this application will not be considered as a binding set of instructions. Its main purpose is to give the user some tools to build a list of assigned works in the order that would minimize the length of a schedule. Used for time management, the mathematical apparatus is designed to provide sufficient grounds for the time-management technology and preferences of a particular user. It does not aim to express analytically a model for the behavior of a particular user.

In the United Institute of Informatics Problems of the National Academy of Sciences of Belarus, currently a software for time management on a modern platform is developed. The approaches presented in the second part of this article will be realized in the system for time management. This system will be useful for small businesses, e.g., for a small team with one leader and one or more subordinates.

Acknowledgements

The research was partially supported by ITHEA International Scientific Society. Research of the second author was partially supported by Belarusian Republican Foundation for Fundamental Research, Project F13M-062.

Bibliography

[Hellsten, 2012] L.-A. M. Hellsten. What Do We Know About Time Management? A Review of the Literature and a Psychometric Critique of Instruments Assessing Time Management. *Time Management, Prof. Todor Stoilov (Ed.), ISBN: 978-953-51-0335-6, InTech, DOI: 10.5772/37248.* Available from: http://www.intechopen.com/books/time-management/what-do-we-know-about-time-management-a-review-of-the-literature-and-a-psychometric-critique-of-inst.

[Jackson, 2009] V.P. Jackson. Time management: a realistic approach. *Journal of the American College of Radiology, 6,* 434 - 436.

[Woolfolk, 1986] A.E. Woolfolk, R.L. Woolfolk. Time management: an experimental investigation. *Journal of School Psychology, 24,* 261 - 275.

[Ho, 2003] B. Ho. Time management of final year undergraduate English projects: supervisees' and the supervisor's coping strategies. *System, 31,* 231 - 245.

[Kelly, 2003] W.E. Kelly. No time to worry: the relationship between worry, time structure, and time management *Personality and Individual Differences, 35,* 1119 - 1126.

[Macan, 2010] T. Macan, J.M. Gibson, J. Cunningham. Will you remember to read this article later when you have time? The relationship between prospective memory and time management. *Personality and Individual Differences, 48,* 725 - 730.

[Keniga, 2013] C.J. Keniga, M. Kleinmann, W. Hehmann. A field test of the quiet hour as a time management technique. *Revue Europienne de Psychologie Applique, 63,* 137 - 145.

[Zampetakis, 2010] L.A. Zampetakis, N. Bouranta, V.S. Moustakis. On the relationship between individual creativity and time management. *Thinking Skills and Creativity, 5,* 23 - 32.

[Ahmad, 2012] N.L. Ahmad, A.N.M. Yusuf, N.D.M. Shobri, S. Wahab. The relationship between time management and job performance in event management. *Social and Behavioral Sciences, 65,* 937 - 941.

[Xu, 2010] J. Xu. Predicting homework time management at the secondary school level: a multilevel analysis. *Learning and Individual Differences, 20,* 34 - 39.

[Hund, 2014] P.M. Hund, J. Dowell, K. Mueller. Representation of time in digital calendars: an argument for a unified, continuous and multi-granular calendar view. *International Journal of Human-Computer Studies, 72,* 1 - 11.

[Darini, 2011] M. Darini, H. Pazhouhesh, f. Moshiri. Relationship between employee's innovation (creativity) time management. *Procedia - Social and Behavioral Sciences, 25,* 201 - 213.

[Nadinloyi, 2013] K.B. Nadinloyi, N. Hajloo, N.S. Garamaleki, H. Sadeghi. The study efficacy of time management training on increase academic time management of students. *Procedia - Social and Behavioral Sciences, 84,* 134 - 138.

[Indreica, 2011] E.-S. Indreica, A.-M. Cazan, C. Truta. Effects of learning styles and time management on academic achievement. *Procedia - Social and Behavioral Sciences, 30,* 1096 - 1102.

[Malita, 2011] L. Malita. Social media time management tools and tips. *Procedia Computer Science, 3,* 747 - 753.

[Sotskov, 2010] Y.N. Sotskov, N.Y. Sotskova, T.-C. Lai, F. Werner. Scheduling under uncertainty: theory and algorithms. Publishing House "Belorusskaya nauka", Minsk, Belarus, 2010.

[Sotskov, 2014] Y.N. Sotskov, F. Werner. A stability approach to sequencing and scheduling under uncertainty. In: Sequencing and Scheduling with Inaccurate Data. Eds. Y.N. Sotskov and F. Werner. Nova Science Publishers, Inc., New York, USA, 2014.

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

[Matsveichuk, 2014] N.M. Matsveichuk, Y.N. Sotskov. A stability approach to two-stage scheduling problems with uncertain processing times. In: Sequencing and Scheduling with Inaccurate Data. Eds. Y.N. Sotskov and F. Werner. Nova Science Publishers, Inc., New York, USA, 2014.

[Sotskov, 2009] Y.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, 50 (3 - 4),* 556 - 573.

[Sotskov, 2010] Y.N. Sotskov, N.G. Egorova, F. Werner. Minimizing total weighted completion time with uncertain data: a stability approach. *Automation and Remote Control, 71 (10),* 2038 - 2057.

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

[Matsveichuk, 2009] N.M. Matsveichuk, Y.N. Sotskov, N.G. Egorova, T.-C. Lai. Schedule execution for two-machine flow-shop with interval processing times. *Mathematical and Computer Modelling, 49 (5 - 6),* 991 - 1011.

[Matsveichuk, 2011] N.M. Matsveichuk, Y.N. Sotskov, F. Werner. The dominance digraph as a solution to the two-machine flow-shop problem with interval processing times. *Optimization, 60 (12),* 1493 - 1517.

[Microsoft Corporation 2009] Microsoft Corporation. Microsoft Application Architecture Guide, 2nd Edition, 10, 2009.