

ROBUST ALGORITHM FOR THE UNCERTAIN SCHEDULING PROBLEM WITH MOVING EXECUTORS

Jerzy Józefczyk

*Institute of Information Science and Engineering, Wrocław University of Technology,
Janiszewski St. 11/17, 50-370 Wrocław, Poland, e-mail: Jerzy.Jozefczyk@pwr.wroc.pl*

Abstract: In the paper the combined routing-scheduling problem which consists in scheduling of tasks on moving executors is considered. The case with non-preemptive and independent tasks, unrelated executors as well as interval processing times to minimize the maximum lateness is investigated. The robust scheduling problem based on the modified relative regret function is formulated. The solution algorithm uses the result which allows to reduce an uncertain problem to a number of deterministic problems. Then the uncertain problem considered can be solved by a deterministic algorithm. The simple numerical example is given. *Copyright © 2005 IFAC*

Keywords: flexible manufacturing systems, scheduling, movement, decision making, uncertainty, robustness, optimization problems

1. INTRODUCTION

Solving task scheduling problems for modern discrete manufacturing systems causes the necessity to take into account new assumptions which make the problems more complex and more difficult. Considering a movement of different elements of manufacturing systems is an example of such a new assumption (Józefczyk, 1997, 2001; Averbakh and Berman, 1999). The movement in discrete manufacturing systems may refer to different objects and may be understood and considered in different ways. One can speak about the movement of plants to be produced, and it is obvious. But one can also consider the movement of executors to perform manufacturing operations on plants located at stationary workstations. A model of the latter type is considered in the paper. A simple scheduling problem with unrelated executors performing a set of non-preemptive, independent tasks to minimize the maximum lateness is investigated. The ready times for all tasks are assumed to be the same. The task scheduling considered are connected also with traveling salesman problems with time windows, see (Dumas, *et al.*, 1995) and (Desrosiers, *et al.*, 1992) for a survey. The main idea of the problem under investigation is the follow-

ing. To perform a *task* an *executor* being the subject performing the tasks should drive-up to a workstation located on a plane or in a space. Therefore, each task consists of two parts: the driving-up to the workstation and the performing of a *job* at the workstation. The generalization of the term *task* leads to the generalization of the execution times which are the main data for every scheduling problem. Then the *execution time* is the sum of the *driving-up time* and the *job performing time*. Such a generalization defines a new scheduling problem.

The paper concerns the generalization of previous investigations towards the non-deterministic case, when the processing times are uncertain. The most popular and widely investigated approach to model the uncertainty in decision making problems is the stochastic approach. It is assumed that some probability distribution over the space of all possible realizations (scenarios) of all random parameters of the problem exists, and the objective is to determine a solution which fulfills a selected probabilistic performance index. The main drawbacks of the stochastic approach are discussed in (Kouvelis and Yu, 1997). The robust approach to manage the uncer-

tainty is also very well known but has no disadvantages of the stochastic approach. It has been used widely also for different discrete optimization problems (e.g. Yu, 1996; Gutierrez and Kouvelis, 1995). Recently, increasing attention has been paid to the robustness approach based on regret functions to decision making – in general or to task scheduling – in particular. This approach leads to the solution that performs well across all scenarios and hedges against the worst of all possible scenarios. In the paper a modified worst-case relative regret function is applied for which the resulting task scheduling algorithm is referred to as the worst-case relative regret task scheduling algorithm, see (Kouvelis and Yu, 1997; Averbakh, 2000).

The investigation of the non-deterministic version of the task scheduling on moving executors is deeply motivated. The uncertainty in the scheduling problem under consideration concerns mainly the execution times, i.e. both the driving-up times and the job performing times. However, in the manufacturing systems both times are often results of other decision making problems, for example they are indirect results of a task or movement control (Józefczyk, 2004a, 2002, 2003). Then the execution times being the parameters of the scheduling problem may be not only incomplete or plausible but also they would be unpredictable. In the latter case it is possible to specify the lower and upper bounds of these times and consequently to determine intervals instead of crisp values of these times like in the deterministic version of the scheduling problem. The main contribution of the paper is based on the result presented in (Averbakh, 2000) that is generalized in terms of an objective function of min-max type. In (Józefczyk, 2004b) the problem which consists in minimization of the makespan has been considered. In the paper the modified version of the relative regret function is proposed. Then the result is applied to the uncertain version of the scheduling problem under consideration which can be reduced to a number of problems without uncertainty.

2. PROBLEMS OF TASK SCHEDULING TO MINIMIZE THE MAXIMUM LATENESS

Let us consider the basic notation. A simple scheduling problem with independent, non-preemptive tasks and unrelated executors as well as the same ready times is the basis for the considerations. However, it deals with the situation when each executor to perform the job should drive-up to the place (workstation), where this job should be done. All assumptions imposed for the classical scheduling problem, i.e. for the problem without movement of executors are valid. For example, each task can be performed only by one executor. Moreover, it is assumed that each executor should begin its movement at a common workstation called a depot for executors (a de-

pot). No distinction between the set of tasks and the set of workstations in notation is introduced. Both sets are denoted as $\mathbf{H} = \{1, 2, \dots, H\}$, where H is the number of tasks to be executed and in the consequence the number of workstations. The depot is distinguished and denoted as $h = H + 1$. Then $\bar{\mathbf{H}} = \mathbf{H} \cup \{H + 1\}$ is the set of workstations with the depot. Analogously, \mathbf{R} and R are the set of executors and the number of executors, respectively. The execution time $\tau_{r,g,h}$ of the task h by the executor r after driving-up from the workstation g is not given a priori. It is the sum of the time the job is performed at the workstation $\tau_{r,h}^I$ and the driving-up time to the workstation from the other workstation $\tau_{r,g,h}^{II}$, i.e.

$\tau_{r,g,h} = \tau_{r,h}^I + \tau_{r,g,h}^{II}$. It is assumed that $\tau_{r,h,h}^{II} = +\infty$, $h = 1, 2, \dots, H$. All execution times form a matrix $\hat{\tau} = [\tau_{r,g,h}]_{\substack{r=1,2,\dots,R, \\ h=1,2,\dots,H}, g=1,2,\dots,H+1}$.

Each task h is characterized not only by the execution time but also by the due-date d_h . Let us assume that the tasks are sorted to fulfill the condition for the due-dates

$$d_g \leq d_h, \text{ for } g, h = 1, 2, \dots, H, g < h. \quad (1)$$

Then one can define additionally the time interval being the period with the beginning at the start of the scheduling procedure and with the end defined by the due-date d_l , where l , $l = 1, 2, \dots, H$ is the index of the time interval l . It is evident that the returns of the executors to the depot have no influence on the value of the maximum lateness and therefore are omitted.

2.1 Optimization problem with moving executors

Let us introduce a matrix of decision variables which constitutes the mapping between the sets of tasks and executors, namely

$$\hat{\alpha} = [\alpha_{r,g,h,l}]_{\substack{r=1,2,\dots,R, \\ h,l=1,2,\dots,H}, g=1,2,\dots,H+1}, \quad (2)$$

where $\alpha_{r,g,h,l} = 1(0)$ if the performance of the task h by the executor r after driving-up from the workstation g is started before the moment when the time interval l ends (otherwise). The proper form of the matrix $\hat{\alpha}$ should guarantee, besides the admissible schedule, the determination of the continuous route for each executor with the beginning at the depot. The following conditions express such requirements

$$\alpha_{r,g,h,l} \in \{0,1\}, \quad r \in \mathbf{R}, g \in \bar{\mathbf{H}}, h \in \mathbf{H}, \quad (3)$$

$$\alpha_{r,h,h,l} = 0, \quad r \in \mathbf{R}, h, l \in \mathbf{H}, \quad (4)$$

$$\sum_{r=1}^R \sum_{g=1}^{H+1} \alpha_{r,g,h,l} = 1, \quad h = 1, 2, \dots, H, l \geq h, \quad (5)$$

$$\sum_{l=1}^H \sum_{g=1}^{H+1} \alpha_{r,g,p,l} = \sum_{l=1}^H \sum_{h=1}^H \alpha_{r,p,h,l}, \quad p=1, 2, \dots, H, \quad (6)$$

$$p \neq \bar{h}^r(H), \quad r=1, 2, \dots, R,$$

where $\bar{h}^r(H)$ is the last task performed by the executor r ,

$$\sum_{l=1}^H \sum_{h=1}^H \alpha_{r,H+1,h,l} = 1, \quad r=1, 2, \dots, R, \quad (7)$$

$$\alpha \in \bar{S}, \quad \text{where}, \quad (8)$$

$$\bar{S} = \{ \alpha : \sum_{l \in \bar{H}_r} \sum_{g \in \bar{H}_r} \sum_{h \in \bar{H}_r} \alpha_{r,g,h,l} \leq \bar{H}_r - 1, \text{ and}$$

$$\bar{H}_r - \text{any non - empty subset of } \bar{H}, \quad r \in R \}.$$

Besides obvious requirements (3) and (4), the others have the following interpretation. The term (5) guarantees that every task is performed by one executor and will be started before its due-date. The continuity of the routes of executors results from (6). The conditions (7) and (8) make sure that each executor starts once from the depot and the routes do not contain sub-cycles, respectively. The constraints (3)–(8) imposed on $\hat{\alpha}$ constitute the set \mathcal{A} of admissible matrices $\hat{\alpha}$. As the performance index the maximum lateness is used

$$Q_L(\bar{\tau}, \hat{\alpha}) = \max_{r=1, 2, \dots, R} \max_{l=1, 2, \dots, H} \left(\sum_{h=1}^{H+1} \sum_{g=1}^H \alpha_{r,g,h,l} \tau_{r,g,h} - d_l \right). \quad (9)$$

To present the performance index in more convenient form for further considerations let us assume the additional notation. Two indices r and l are replaced by one index

$$k = l + H(r-1), \quad (10)$$

where the index l changes from 1 to H for every $r=1, 2, \dots, R$, i.e. for fixed r the increase of l involves the same increase of k . The elements of the matrices $\bar{\tau}$ and $\hat{\alpha}$ for every k are placed into vectors

$$\tau_k = [-d_l, \tau_{r,1,1}, \tau_{r,1,2}, \dots, \tau_{r,1,H}, \tau_{r,2,1}, \tau_{r,2,2}, \dots, \tau_{r,2,H}, \dots, \tau_{r,H+1,1}, \tau_{r,H+1,2}, \dots, \tau_{r,H+1,H}], \quad (11)$$

$$\alpha_k \triangleq \alpha_{r,l} = [1, \alpha_{k,1,1}, \alpha_{k,1,2}, \dots, \alpha_{k,1,H}, \alpha_{k,2,1}, \alpha_{k,2,2}, \dots, \alpha_{k,2,H}, \dots, \alpha_{k,H+1,1}, \alpha_{k,H+1,2}, \dots, \alpha_{k,H+1,H}], \quad (12)$$

where $\alpha_{k,g,h} \triangleq \alpha_{r,g,h,l}$ which constitute a new matrices $\tau = [\tau_1^T, \tau_2^T, \dots, \tau_k^T, \dots, \tau_{RH}^T]$ and $\alpha = [\alpha_1^T, \alpha_2^T, \dots, \alpha_k^T, \dots, \alpha_{RH}^T]$. It is easy to see that in the matrix τ occur only R columns which are different with respect to the execution times and that correspond to R sections of H the same columns, i.e. in each of the first H columns besides of $-d_l$ the

execution times $\tau_{1,g,h}$ appear, in each of the next H columns besides of $-d_l$ the execution times $\tau_{2,g,h}$ take place and so on. For $R=2$ and $H=3$ the matrix τ has the following form

$$\tau = \begin{bmatrix} -d_1 & -d_2 & -d_3 & -d_1 & -d_2 & -d_3 \\ \tau_{1,1,1} & \tau_{1,1,1} & \tau_{1,1,1} & \tau_{2,1,1} & \tau_{2,1,1} & \tau_{2,1,1} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \tau_{1,4,3} & \tau_{1,4,3} & \tau_{1,4,3} & \tau_{2,4,3} & \tau_{2,4,3} & \tau_{2,4,3} \end{bmatrix}.$$

Then the performance index can be expressed as

$$Q_L(\tau, \alpha) = \max_{k=1, 2, \dots, RH} \tau_k \alpha_k^T. \quad (13)$$

Now we can formulate the scheduling problem with moving executors as a discrete optimization problem $\mathbf{P}(\tau)$ which consists in the minimization of $Q_L(\tau, \alpha)$ with respect to the admissible α for fixed value of τ , i.e. $\min_{\alpha \in \mathcal{A}} Q_L(\tau, \alpha)$, and $Q_L^*(\tau) \triangleq \min_{\alpha \in \mathcal{A}} Q(\tau, \alpha)$.

2.2 Optimization problem without moving executors

When the movement of executors is not considered the corresponding scheduling problem is the special case of the problem formulated in the previous subsection. It consists in minimization of the maximum lateness

$$Q'_L(\tau', \alpha') = \max_{k=1, 2, \dots, RH} \tau'_k \alpha'_k{}^T, \quad (14)$$

where $\tau'_k = [-d_l, \tau_{r,1}, \tau_{r,2}, \dots, \tau_{r,H}]$ and $\alpha'_k = [1, \alpha'_{k,1}, \alpha'_{k,2}, \dots, \alpha'_{k,H}]$ with respect to the admissible matrix $\alpha' = [\alpha'_1{}^T, \alpha'_2{}^T, \dots, \alpha'_{RH}{}^T]$, where $\alpha'_{k,h} \triangleq \alpha'_{r,h,l} = 1(0)$ if task h performed by executor r starts before the end of interval l (otherwise). The constraints

$$\sum_{r=1}^R \alpha'_{r,h,l} = 1, \quad h=1, 2, \dots, H, \quad l \geq h \quad (15)$$

constitute the set \mathcal{A}' of admissible matrices α' .

2.3 Version with moving executors and with interval processing times

In many cases the exact values of the execution times $\tau_{r,g,h}$ or their components are not known. Let us assume that $\tau_{r,g,h} \in [\underline{\tau}_{r,g,h}, \bar{\tau}_{r,g,h}]$, where the values of $\underline{\tau}_{r,g,h}$ and $\bar{\tau}_{r,g,h}$ are given. Then each realization of the matrix τ of the execution times $\tau_{r,g,h}$ being the parameters of the scheduling problem $\mathbf{P}(\tau)$ is called a scenario. The set \mathcal{T} of all scenarios

is the Cartesian product of all intervals $[\underline{\tau}_{r,g,h}, \bar{\tau}_{r,g,h}]$. The worst case approach is proposed to cope with such an uncertainty. The objective is to find the matrix α that performs well for all scenarios in the sense of the maximum lateness (13). The robust optimization approach can be applied, e.g. (Rosenblatt and Lee, 1987; Kouvelis and Yu, 1997; Averbakh, 2000; Yager, 2004). The uncertainty in decisions due to the uncertainty in parameters τ_k can be evaluated in different ways. Because the value of $Q_L^*(\tau)$ can be negative for further considerations the modified relative regret function is used being the ratio

$$\left| \frac{Q_L(\tau, \alpha) - Q_L^*(\tau)}{Q_L^*(\tau)} \right| = \frac{Q_L(\tau, \alpha) - Q_L^*(\tau)}{|Q_L^*(\tau)|}.$$

The function fits with the proposition given in (Kouvelis and Yu, 1997; Averbakh, 2000) for non-negative values of the function corresponding to $Q_L^*(\tau)$ and preserves all their properties for the negative values. Then an uncertain optimization problem referred to as **UP** consists in the minimization of

$$z_L(\alpha) = \max_{\tau \in T} \frac{Q_L(\tau, \alpha) - Q_L^*(\tau)}{|Q_L^*(\tau)|} \quad (16)$$

with respect to $\alpha \in \mathcal{A}$, i.e.

$$\min_{\alpha \in \mathcal{A}} z_L(\alpha) = \min_{\alpha \in \mathcal{A}} \max_{\tau \in T} \frac{Q_L(\tau, \alpha) - Q_L^*(\tau)}{|Q_L^*(\tau)|}. \quad (17)$$

As the result the robust optimal matrix $\tilde{\alpha}$ expressing the robust scheduling of tasks as well as the optimal value of the worst-case relative regret function $\tilde{z}_L \triangleq z_L(\tilde{\alpha})$ are obtained. For the notational convenience it is assumed that any ratio a/b is equal to 0 for $b=0$ and $a \neq 0$ and to $+\infty$ for $a=b=0$. Now we give an interpretation of the problem **UP** and the performance index (16). If the inequality

$$(Q_L(\tau, \alpha) - Q_L^*(\tau)) / Q_L^*(\tau) < \varepsilon \quad (18)$$

holds for $\varepsilon > 0$ then $\alpha \in \mathcal{A}$ is the relative ε -optimal solution of **P**(τ). Moreover, let

$\Gamma_\varepsilon(\tau) = \{\alpha \in \mathcal{A} : (Q_L(\tau, \alpha) - Q_L^*(\tau)) / |Q_L^*(\tau)| < \varepsilon\}$ be the set of all relative ε -optimal solutions of **P**(τ). For given $\varepsilon > 0$ it is crucial to determine such solutions for all τ which fulfill the condition

$$\alpha \in \bigcap_{\tau \in T} \Gamma_\varepsilon(\tau). \quad (19)$$

The existence of such α depends on ε . For small ε the solution can not exist because the admissible solution for the determined τ may be unacceptable

for the other one. We denote by $z_L(\tilde{\alpha})$ the least value of ε for which (19) is fulfilled. If the inequality (18) is true for the worst case, i.e.

$\max_{\tau \in T} (Q_L(\tau, \alpha) - Q_L^*(\tau)) / |Q_L^*(\tau)| < \varepsilon$ it is also true

for every τ . To enable the fulfilling of the inequality for possibly the least $\varepsilon > 0$ it is necessary to minimize it, i.e. to solve the problem (17). Then the value \tilde{z}_L is a measure of the uncertainty and finally the problem **UP** consists in the determination of such a $\alpha \in \mathcal{A}$ that the relative deviation from $|Q_L^*(\tau)|$, the worst with respect to τ , called the worst-case relative regret function, is the least.

3. WORST-CASE RELATIVE REGRET SOLUTION ALGORITHM

Now we consider a new optimization problem **UP'** equivalent to **UP**. Let τ^k be the matrix τ with the execution times $\tau_{r,g,h}$ belonging to the vector τ_k equal to $\bar{\tau}_{r,g,h}$. The other execution times of τ are equal to $\underline{\tau}_{r,g,h}$. The set of all matrices τ^k we denote as $T' = \{\tau^k, k = 1, 2, \dots, RH\}$. Then **UP'** deals with the minimization of the maximum value of functions $z_{L,k}(\alpha)$ defined as

$$z_{L,k}(\alpha) = \frac{\bar{\tau}_k \alpha_k^T - Q_L^*(\tau^k)}{|Q_L^*(\tau^k)|}, \quad (20)$$

where

$\bar{\tau}_k = [-d_l, \bar{\tau}_{r,1,1}, \bar{\tau}_{r,1,2}, \dots, \bar{\tau}_{r,1,H}, \bar{\tau}_{r,2,1}, \bar{\tau}_{r,2,2}, \dots, \bar{\tau}_{r,2,H}, \dots, \bar{\tau}_{r,H+1,1}, \bar{\tau}_{r,H+1,2}, \dots, \bar{\tau}_{r,H+1,H}]$, with respect to admissible α , i.e.

$$\begin{aligned} & \min_{\alpha \in \mathcal{A}} \max_{k=1, 2, \dots, RH} z_{L,k}(\alpha) \\ & = \min_{\alpha \in \mathcal{A}} \max_{k=1, 2, \dots, RH} \frac{\bar{\tau}_k \alpha_k^T - Q_L^*(\tau^k)}{|Q_L^*(\tau^k)|}. \end{aligned} \quad (21)$$

The lemma is true.

Lemma. For any $\alpha \in \mathcal{A}$

$$z_L(\alpha) = \max_{k=1, 2, \dots, RH} z_{L,k}(\alpha). \quad (22)$$

Proof of Lemma. Let $\tilde{\tau}$ be the matrix for an arbitrary $\alpha \in \mathcal{A}$ for which the inequality

$$\frac{Q_L(\tilde{\tau}, \alpha) - Q_L^*(\tilde{\tau})}{|Q_L^*(\tilde{\tau})|} \geq \frac{Q_L(\tau, \alpha) - Q_L^*(\tau)}{|Q_L^*(\tau)|} \quad (23)$$

holds for $\tau \in T$ and let $\tilde{\alpha}$ be the solution of **P**($\tilde{\tau}$) for fixed $\tilde{\tau}$. Moreover, let us denote by s such an index that $\tau_s \alpha_s^T \geq \tau_k \alpha_k^T$, $k = 1, 2, \dots, RH$. To notice

that the value of the left hand side in (23) can not decline, if in $\tilde{\tau}$ we increase $\tilde{\tau}_s$ and decrease $\tilde{\tau}_k$ for $k \neq s$, it is enough to consider the equalities

$$\begin{aligned} & (Q_L(\tilde{\tau}, \alpha) - Q_L^*(\tilde{\tau})) / |Q_L^*(\tilde{\tau})| \quad (24) \\ &= \frac{\max_{k=1,2,\dots,RH} \tilde{\tau}_k \alpha_k^T - \min_{\alpha \in \mathcal{A}} \max_{k=1,2,\dots,RH} \tilde{\tau}_k \alpha_k^T}{\left| \min_{\alpha \in \mathcal{A}} \max_{k=1,2,\dots,RH} \tilde{\tau}_k \alpha_k^T \right|} \\ &= \frac{\tilde{\tau}_s \alpha_s^T - \min_{\alpha \in \mathcal{A}} \max_{k=1,2,\dots,RH} \tilde{\tau}_k \alpha_k^T}{\left| \min_{\alpha \in \mathcal{A}} \max_{k=1,2,\dots,RH} \tilde{\tau}_k \alpha_k^T \right|}. \end{aligned}$$

On the other hand the value of the left hand side in (23) can not increase because of the definition of $\tilde{\tau}$. Using (16) and (24) the equation (22) can be immediately justified

$$\begin{aligned} z_L(\alpha) &= \max_{\tau \in T} \frac{Q_L(\tau, \alpha) - Q_L^*(\tau)}{|Q_L^*(\tau)|} \\ &= \frac{Q_L(\tilde{\tau}, \alpha) - Q_L^*(\tilde{\tau})}{|Q_L^*(\tilde{\tau})|} = \frac{Q_L(\tau^s, \alpha) - Q_L^*(\tau^s)}{|Q_L^*(\tau^s)|} \\ &= \frac{\tilde{\tau}_s \alpha_s^T - Q_L^*(\tau^s)}{|Q_L^*(\tau^s)|} = z_{L,s}(\alpha). \end{aligned}$$

Finally, from $z_L(\alpha) \geq z_{L,k}(\alpha)$ for $k \neq s$ results

$$z_L(\alpha) = \max_{k=1,2,\dots,RH} z_{L,k}(\alpha). \quad \blacksquare$$

In the consequence, for any $\alpha \in \mathcal{A}$ there is the matrix $\tilde{\tau}$ for which the inequality (23) holds. Therefore, the set of optimal solutions for \mathbf{UP}' doesn't change if we replace T with T' and hence the problems \mathbf{UP} and \mathbf{UP}' are equivalent.

Let us introduce the sets of matrices $\tau'' = [\tau_1'', \tau_2'', \dots, \tau_{RH}'']$,

$$\begin{aligned} \tau_k'' &= \left[\frac{-d_l}{|Q_L^*(\tau^k)|}, \tau_{r,1,1}'', \tau_{r,1,2}'', \dots, \tau_{r,1,H}'', \tau_{r,2,1}'', \tau_{r,2,2}'', \dots, \right. \\ & \left. \tau_{r,2,H}'', \dots, \tau_{r,H+1,1}'', \tau_{r,H+1,2}'', \dots, \tau_{r,H+1,H}'' \right], \text{ where} \\ \tau_{r,g,h}'' &= \frac{\bar{\tau}_{r,g,h}}{|Q_L^*(\tau^k)|} = \frac{\bar{\tau}_{r,g,h}}{|Q_L^*(\tau^r)|} \quad (25) \end{aligned}$$

and k is defined in (10). The last equation in (25) results from the definition of matrix τ . The following theorem is the basis for the determination of the worst-case relative regret task scheduling algorithm.

Theorem. The sets of optimal solutions of $\mathbf{P}(\tau'')$ and \mathbf{UP}' are the same, i.e. to solve \mathbf{UP}' it is enough to solve $\mathbf{P}(\tau'')$.

Proof of Theorem. Notice that

$$z_{L,k}(\alpha) = \frac{\bar{\tau}_k \alpha_k^T - Q_L^*(\tau^k)}{|Q_L^*(\tau^k)|} = \frac{\bar{\tau}_k \alpha_k^T}{|Q_L^*(\tau^k)|} - 1 = \tau_k'' \alpha_k^T - 1.$$

Then, the solution of \mathbf{UP}' can be transformed

$$\begin{aligned} \min_{\alpha \in \mathcal{A}} \max_{k=1,2,\dots,RH} z_{L,k}(\alpha) &= \min_{\alpha \in \mathcal{A}} \max_{k=1,2,\dots,RH} (\tau_k'' \alpha_k^T - 1) \\ &= \min_{\alpha \in \mathcal{A}} \max_{k=1,2,\dots,RH} \tau_k'' \alpha_k^T - 1 = \min_{\alpha \in \mathcal{A}} Q_L(\tau'', \alpha) - 1 \\ &= Q_L^*(\tau'') - 1. \quad \blacksquare \end{aligned}$$

Thus, to solve \mathbf{UP}' it is enough to solve $\mathbf{P}(\tau'')$ because the values of elements of τ'' are deterministic. It is necessary to stress that in general $\mathbf{P}(\tau'')$ is not the task scheduling problem defined in Subsection 2.1 because $d_l / |Q_L^*(\tau^k)|$ doesn't fit the definition of due-dates. Now, we are able to present the solution algorithm for \mathbf{UP}' .

1. Solve the problems $\mathbf{P}(\tau^k)$, $k=1,2,\dots,RH$ to obtain the values $|Q_L^*(\tau^k)|$.

2. Solve the deterministic decision making problem related to $\mathbf{P}(\tau)$, i.e. with new execution times defined in (25), the due-dates equal to $d_l / |Q_L^*(\tau^k)|$ and the indices l, r, k connected according to (10).

Because of the structure of the matrix τ in Step 1 it is enough to solve problems $\mathbf{P}(\tau^r)$, $r=1,2,\dots,R$, any time with the different column of τ containing the maximum values of $\tau_{r,g,h}$ and the other execution times equal to $\tau_{p,g,h}$, $p \neq r$. It is easy to formulate the remark that for identical executors, in Step 2 of the algorithm the deterministic task scheduling problem considered is solved with the execution times given in (25) and the due-dates $d_l / |Q_L^*(\tau^k)|$.

4. NUMERICAL EXAMPLE

To illustrate the solution algorithm the elementary numerical example is presented. It concerns the version with stationary identical executors. The properties as well as the solution algorithm for the version with moving executors can be applied analogously. Let $R=2$, $H=6$ and the interval execution times as well as the due-dates are given in Table 1. During the first step of the algorithm two task scheduling problems are solved with the execution times presented in Table 2 and Table 3 for $r=1$ and $r=2$, respectively. As the result the following values of the maximum lateness are obtained $Q_L^*(\tau^1) = 1$,

Table 1 Data for the numerical example

h	1	2	3	4	5	6
$\underline{\tau}_{1,h}, \bar{\tau}_{1,h}$	4,6	1,3	6,8	2,10	5,7	2,6
$\underline{\tau}_{2,h}, \bar{\tau}_{2,h}$	1,7	1,5	3,7	4,12	1,3	4,10
d_h	4	5	6	7	8	9

Table 2 Data for the problem $\mathbf{P}(\tau^1)$

h	1	2	3	4	5	6
$\bar{\tau}_{1,h}$	6	3	8	10	7	6
$\underline{\tau}_{2,h}$	1	1	3	4	1	4

Table 3 Data for the problem $\mathbf{P}(\tau^2)$

h	1	2	3	4	5	6
$\underline{\tau}_{1,h}$	4	1	6	2	5	2
$\bar{\tau}_{2,h}$	7	5	7	12	3	10

Table 4 Data for Step 2 of the algorithm

h	1	2	3	4	5	6
$\bar{\tau}_{1,h}''$	6	3	8	10	7	6
$d_h / Q_L^*(\tau^1) $	4	5	6	7	8	9
$\bar{\tau}_{2,h}''$	3.5	2.5	3.5	6	1.5	5
$d_h / Q_L^*(\tau^2) $	2	2.5	3	3.5	4	4.5

$Q_L^*(\tau^2) = 2$. The algorithm in Step 2 uses the data given in Table 4. As the final result the value $\tilde{z}_L = 7$ as well as the matrix $\tilde{\alpha}$ which corresponds to the schedule (1, 2, 6) and (3, 4, 5) for $r = 1$ and $r = 2$, respectively are determined.

5. CONCLUSIONS

It is obvious that taking into account the uncertainty of the processing times additionally increases the complexity of the task scheduling problem considered. For the case of the interval times the result given in (Averbakh, 2000) was generalized and used for the task scheduling problem under consideration. The algorithm proposed assumes the information of the exact solution of the scheduling problem when the NP-hard optimization problems should be solved. Therefore it is crucial to extend in further considerations the approach for the approximate algorithms and in the consequence to enable solving large instances of the scheduling problem. However it requires a new definition of the relative regret function.

ACKNOWLEDGEMENTS

The research was supported in 2005-2007 by the State Committee for Scientific Research under Grant No. 3 T11A 031 28.

REFERENCES

- Averbakh, I., O. Berman (1999). A simple heuristic for m -machine flow-shop and its applications in routing-scheduling problems. *Operations Research*, **47**, 165-170, 1999.
- Averbakh, I. (2000). Minimax regret solutions for minimax optimization problems with uncertainty. *Operations Research Letters*, **27**, 57-65.
- Daniels, R.L., P. Kouvelis (1995). Robust scheduling to hedge against processing time uncertainty in single-stage production. *Management Science*, **41**, 363-376.
- Desrosiers, J., Y. Dumas, M.M. Solomon and F. Soumis (1992). Time constrained routing and scheduling. *Report G-92-42*, GERAD, Montreal.
- Dumas, Y., J. Desrosiers, E. Gelinas and M.M. Solomon (1995). An optimal algorithm for the traveling salesman problem with time windows. *Operations Research*, **43**, 367-371.
- Gutierrez, G.J., P. Kouvelis (1995). A robustness approach to international sourcing. *Annals of Operations Research*, **59**, 165-193.
- Józefczyk, J. (1997). An algorithm for scheduling tasks on moving executors in complex operation systems. In: *Proceedings of 1st IFAC Workshop on Manufacturing Systems MIM'97*, Vienna, Austria, 139-144.
- Józefczyk, J. (2001). Scheduling tasks on moving executors to minimize the maximum lateness. *European Journal of Operational Research*, **131**, 171-187.
- Józefczyk, J. (2002). Application of knowledge-based pattern recognition to movement control of a group of vehicles. *International Journal of Knowledge Based Intelligent Engineering Systems*, **6**, 192-198.
- Józefczyk, J. (2003). Knowledge-based two-level control of a group of mobile executors. *Integrated Computer-Aided Engineering*, **10**, 191-202.
- Józefczyk, J. (2004a). Decision-making algorithms in two-level complex operation system. *Decision Support Systems*, **38**, 171-182.
- Józefczyk, J. (2004b). Robust algorithm for task scheduling on moving executors with uncertain processing times. In: *Proceedings of the 15th Mini EURO Conf. on Managing Uncertainty in Decision Support Models*, Coimbra, Portugal [cd-rom].
- Kouvelis, P., G. Yu (1997). *Robust discrete optimization and its applications*. Kluwer Academic Publishers, Boston.
- Rosenblatt, M.J., H.L. Lee (1987). A robustness approach to facilities design, *Int. J. Production Res.*, **25**, 479-486.
- Yager, R.R. (2004). Decision making using minimization of regret. *Int. J. of Approximate Reasoning*, **36**, 109-128.
- Yu, G. (1996). On the 0-1 knapsack problem with robust optimization applications. *Operations Research*, **44**, 407-415.