Les cahiers Leibniz

Single-machine production and maintenance: a unified approach

Gerd Finke, Marie-Laure Espinouse, Ahmed Gara-Ali, Vincent Jost, Julien Moncel

Laboratoire G-SCOP
46 av. Félix Viallet, 38000 GRENOBLE, France
ISSN : 1298-020X

n° 205

July 2013

Site internet : http://www.g-scop.inpg.fr/CahiersLeibniz/

Single-machine production and maintenance: a unified approach

G. Finke, M.-L. Espinouse, A. Gara-Ali,
V. Jost, J. Moncel

Abstract

We are presenting a general unifying framework for production-maintenance systems on a single machine. Several performance criteria are included as well as most of the maintenance systems that are proposed in the literature. No particular analytical functions are required and no new algorithms have to be developed. One may simply utilize standard software for min-cost network flow problems.

1 Introduction

There is a vast literature on single-machine production systems with learning and deterioration effects. In particular in manual or semi-automatic processes, the processing times may be shortened since the operators can use their past experience. On the other side, the production may slow down with the deterioration of the tools that are to be used on the production site.

In the more recent literature, maintenance periods are to be inserted into the production system to restore the initial processing conditions. Also such maintenance periods may incorporate a learning effect (the maintenance crew may learn from their previous maintenance) or they may deteriorate with time (a postponed maintenance may take longer).

We shall in this note concentrate on so called positional changes of the processing times. A feasible schedule is given by the job sequence on a single machine. Thus the processing time of job j depends on its position or rank r in the sequence and may eventually also be influenced by its location with respect to the maintenance intervals.

In most articles, specific analytical functions for the processing times are proposed, which will always be restrictive in some way. We refer as reference on this topic to the recent articles [6], [11], [13] and [14] with their listed bibliography on production-maintenance systems. In particular in [11] more general positional functions are considered.

However, all these models are not as general as they could be. In fact, there are no explicit analytical functions at all required to capture the tendencies, *i.e.* learning or deterioration. We shall present an even more general unifying framework. No algorithm will have to be adapted to this production-maintenance environment. One may use simply any commercial linear programming software, or even better min-cost network flow codes that are often included as an option in the software (for instance in CPLEX). More precisely, all the required network flow problems will correspond to min-cost matchings in complete bipartite graphs. These problems belong to the most thoroughly studied problems in discrete mathematics (see for instance [1] and [12]). We shall just indicate the existing complexity results to get an idea about the best running times for all solution procedures that are proposed.

2 Positional production systems

We consider single-machine problems for which the processing time of job j $(j = 1, \ldots, n)$ when scheduled in rank r $(r = 1, \ldots, n)$ is given by $p(j, r)$. The processing times are *deteriorating* if for all j

$$p(j, 1) \leq p(j, 2) \leq ... \leq p(j, n)$$

and we have a *learning effect* if

$$p(j, 1) \geq p(j, 2) \geq ... \geq p(j, n).$$

The functions from the literature include for instance the following: $p_j r^a$, $p_j r^{a_j}$ and $p_j \gamma^{r-1}$. In [11] a more general form is given: $p_j g(r)$ and $p_j g_j(r)$ where g and g_j are arbitrary nondecreasing functions. We shall not assume any analytical form, but rather require that in the given production system one can evaluate the processing times $p(j, r)$ and hence obtain an nxn-table of values. Since one may have at first a learning effect and then after some time a deterioration because of tool wear, we shall not require any monotonicity of the values $p(j, r)$.

Observation 1 *Let an nxn-table of values* $\{f(j, r) : j, r = 1, \ldots, n\}$ *be given, where* f *is an arbitrary function. Solve the assignment problem based on the matrix* $\{f(j, r))\}$*. An optimal solution selects for each row* j *exactly one column* $\pi(j) = r$ *so that the sum* $\sum_{j=1}^{n} f(j, \pi(j))$ *is minimized. The optimal permutation* π *can be obtained in* $O(n^3)$ *time (see [5]).*

It is difficult to tell who the first was to apply this result to positional production systems. It seems that E.L. Lawler made this observation in the 1970's. This result appeared then systematically in the scheduling literature from 1999 on ([2]).

Now let position-dependent processing times be given $p(j,r)$. Denote by C_j the completion time of job j in some sequence π. Using Observation 1 and setting $p(j,r) = f(j,r)$, one obtains directly the optimal job sequence π for the makespan problem $1/p(j,r)/C_{max}$.

However, the makespan is just one possible criterion that fits into this scheme. Two more single-machine scheduling problems can be solved in the same way (see for instance [10]). The first is the problem of minimizing the *total completion time*, $1/p(j,r)/\sum C_j$, where one simply has to solve the assignment problem for the function $f(j,r) = (n-r+1)p(j,r)$. Another criterion is the *total absolute difference in completion times*, in short $TADC$, defined as $TADC = \sum_{i<j} |C_i - C_j|$. The problem $1/p(j,r)/TADC$ is solved as assignment problem based on the function $f(j,r) = p(j,r)\{\sum_{i\geq r}(2i - (n+1))\}$.

Let γ indicate any of these three performance criteria, *i.e.*

$$\gamma \in \{C_{max}, \sum C_j, TADC\}.$$

Proposition 1 *The single-machine scheduling problems*

$$1/p(j,r)/\gamma$$

can be modeled as assignment problems. Hence they are solved in $O(n^3)$ time, where n is the number of jobs.

The processing times $p(j,r)$ are *decomposable* if they are of the form $p(j,r) = p_j g(r)$. The position-dependent processing times in the literature are usually decomposable.

Observation 2 *([7]) Let two vectors of numbers $(x_1, x_2, ..., x_n)$ and $(y_1, y_2, ..., y_n)$ be given. Then their scalar product is minimized if the components of one of the vectors are taken in nondecreasing order and the components of the other vector are taken in nonincreasing order.*

This well-known result was not used for a long time in the area of decomposable positional processing times. It seems that this property has been applied systematically only since 2009 ([6], see also [10], [11]). Thus in the past the proofs for various functions $g(r)$ were in essence repeating the exchange argument of [7]. We get immediately the following result without any assumption on the functions $g(r)$.

Proposition 2 *The single-machine scheduling problems with decomposable processing times:*

$$1/p_j g(r)/\gamma$$

can be solved by sorting in $O(nlogn)$ time, where n is the number of jobs.

3 Insertion of maintenance periods

In an environment with positional processing times $p(j,r)$ that are now assumed to be deteriorating, *i.e.* $p(j,1) \le p(j,2) \le ... \le p(j,n)$, one has to insert from time to time maintenance periods to return to good processing conditions.

Depending on the availability of maintenance crews on the production line, a certain total number k of maintenance periods will be allocated. Typically the best number k of periods that minimizes the given performance criterion has to be determined.

Different maintenance models have been proposed. The duration t_i of all maintenance periods may be constant, *i.e.* $t_i = \beta$ for all periods $i = 1, \ldots, k$, where β is a given number. This model reflects the fact that very often the maintenance consists of a fixed number of prescribed routine actions that have to be performed.

The durations may be constant but different for the maintenance periods, *i.e.* $t_i = \beta_i$ for period i. Here different maintenance crews may intervene, who have different skills and experience to carry out the routine actions.

In [13] the durations are of the form $t_i = t_0 i^b$, where $b < 0$ indicates a learning factor. The maintenance crew may learn by repeating the same maintenance work and reduce the time to do the routine actions.

Without specifying any particular form, we know that in all these models the total maintenance time $T_k = \sum_{i=1}^{k} \beta_i$ with k periods is constant. Hence we can measure the performance of the production process as the trade-off between the maintenance time T_k and the total processing time on the machine.

We shall specify a closed maintenance system: as a user we start the processing with a well maintained machine and at the end we hand back the machine in perfect condition, *i.e.* we terminate with a maintenance period. In an open system (no maintenance at the end) we may get an undesirable non symmetric behavior, accumulating the long difficult jobs at the end since no further maintenance is required. We also assume that the job sequence between two maintenance periods may be void for a certain k, which just tells us that this k is not the optimal number of periods. Therefore, for n

given jobs, we consider $k = 1, \ldots, n$ possible maintenance periods.

We denote by MP maintenance systems for which the maintenance times T_k are given and independent of processing times. The number k is variable and the best k has to be determined. The corresponding scheduling problem is given as $1/p(j,r), MP/\gamma$.

There is a further interesting maintenance model, denoted $MP(\tau)$, for which the length of a maintenance period depends on the time elapsed since the last maintenance [13]. Here the rational is that the processing conditions deteriorate and more maintenance is required the longer one is waiting. In detail: let τ_i be the total processing time between maintenance periods $i-1$ and i (setting τ_1 equal to the start time of the first period). Then the length of period i is given in the form $t_i = \alpha\tau_i + \beta_i$ and the corresponding scheduling problem is defined as $1/p(j,r), MP(\tau)/\gamma$.

We generalize further and introduce maintenance periods, whose durations are weighted by factors $w(j,r)$. The idea is the following. We combine the jobs processed in some group i with the subsequent maintenance period i. Whenever job j is processed in rank r within group i, the subsequent maintenance period i requires the time $w(j,r)p(j,r)$ to restore the initial processing conditions, in addition to some constant time β_i. These weights $w(j,r)$ are not decreasing with r, *i.e.* $w(j,1) \leq w(j,2) \leq ... \leq w(j,n)$ for all jobs j. Then the duration of period i is equal to

$$t_i = \sum_{j \in group(i)} w(j,r)p(j,r) + \beta_i.$$

We call such weighted maintenance systems $MP(w)$ and the corresponding scheduling problem is defined as $1/p(j,r), MP(w)/\gamma$. The system $MP(w)$ transforms to $MP(\tau)$ by setting $w(j,r) = \alpha$ and to MP by setting $w(j,r) = 0$.

We can keep track of the variable part $w(j,r)p(j,r)$ of the system $MP(w)$ by simply changing the positional entries $f(j,r)$ to $f(j,r) + w(j,r)p(j,r)$ and reduce the duration of period i to the constant time β_i to get the same result. After having obtained an optimal solution, one can shift back the times $w(j,r)p(j,r)$ in group i to the maintenance period i to get the total maintenance time t_i. With this transformation we have reduced the maintenance system $MP(w)$ to the system $MP(\tau)$ and to the system MP.

Lemma 1 *Let a production-maintenance system be given with processing times $p(j,r)$ and known positional entries $f(j,r)$. Then the maintenance systems $MP(w)$, $MP(\tau)$ and MP are equivalent by replacing the entries $f(j,r)$ by*

$$f(j,r) + w(j,r)p(j,r).$$

Remember that for C_{max} we have $p(j,r) = f(j,r)$. But when dealing with $\sum C_j$ and $TADC$ the positional entries $f(j,r)$ also depend on the group length l. For $\sum C_j$ we have for instance $f(j,r) = (l-r+1)p(j,r)$ if group i contains l jobs. In the following we suppose that the transformation above is done whenever possible, and we only have to deal with maintenance systems, called $\mathcal{M}$, that are of the type MP having only periods with constant durations β_i.

3.1 Balanced schedules

A schedule with k maintenance periods is called *balanced* if the number of jobs in each of the k groups of jobs (*i.e.* the k sets of jobs processed between consecutive maintenance periods) differs by at most one (see [8]).

Let us concentrate first on the makespan C_{max}. In this case, there is an optimal schedule that is balanced for all maintenance systems $\mathcal{M}$. This is valid since having two groups of jobs between maintenance periods with $l+2$ jobs and the other with l or fewer jobs, one can shift the job j in rank $l+2$ to the rank $l+1$ in the other group. We know that both the processing times $p(j,r)$ and the weights $w(j,r)$ are deteriorating with r. Hence the total processing time for maintenance MP is reduced since the change is

$$p(j,l+2) - p(j,l+1) \geq 0.$$

Consider now the maintenance systems $MP(\tau)$ and $MP(w)$. The total processing time is again reduced by this exchange since

$$(1 + w(j,l+2))p(j,l+2) - (1 + w(j,l+1))p(j,l+1) \geq 0.$$

In the simplest reported case, one has the maintenance system MP and the function $p(j,r) = p_j g(r)$ where $g(1) \leq g(2) \leq ... \leq g(n)$. One sorts the p_j in nonincreasing order in $O(nlogn)$ time. Then one fills k groups, where k is a given number of periods, starting with the k jobs having the smallest p_j in rank 1, then the next k jobs with the smallest remaining p_j in rank 2 assigned in any order to the groups, and so on, until all k groups are filled to the rank $l-1$ where $l = \lceil \frac{n}{k} \rceil$. Then the remaining $n-(l-1)k$ jobs are placed in rank l in any $n-(l-1)k$ groups, all other $lk-n$ groups remain filled till the rank $l-1$. Evaluating the objective function for all k with $1 \leq k \leq n$ to find its minimum takes $O(n^2)$ time. Hence $O(n^2)$ is the complexity to solve problem $1/p_j g(r), MP/C_{max}$ ([8] and [11]).

Observation 3 *(See for instance [1] P. 153) A balanced nxm transportation problem with total supply being equal to the total demand, i.e.* $M = \sum_{j=1}^{n} a_j = \sum_{i=1}^{m} b_i$*, can be solved in* $O(M(n+m)^2)$ *time.*

For the makespan C_{max} and for each k we define the following transportation problem. Let n be the number of jobs and set $a_j = 1$ for $j = 1, \ldots, n$. Set $m = l = \lceil \frac{n}{k} \rceil$ and define $b_i = k$ for $i = 1, \ldots, l-1$ and $b_l = n - (l-1)k$. Then M=n and the corresponding transportation problem is solved in $O(n(n+l)^2) = O(n^3)$ time for any k to return the makespan with k maintenance periods. The allocation of jobs to the k groups has just to respect the rank, but is otherwise arbitrary. One has many alternate optimal solutions by exchanging the processing terms of the same rank from one group to another. This is also changing for the maintenance systems $MP(w)$ and $MP(\tau)$ the times t_i since the variable part is modified as a result of the exchange step. We find the best k by solving n transportation problems in $O(n^4)$ time.

Proposition 3 *The problems*

$$1/p(j,r), \mathcal{M}/C_{max}$$

possess an optimal schedule that is balanced, and this schedule can be found in $O(n^4)$ time by solving a series of transportation problems.

Let us turn to the other two criteria $\sum C_j$ and $TADC$. The balanced schedules are not necessarily optimal. Take for instance the criterion $\sum C_j$ with $n = 4$ jobs. Three jobs are identical: $p(j,r) = 1, 2, 3, 4$ $(j = 1, 2, 3)$ and $p(4,r) = 10, 20, 20, 20$. All maintenance periods take 10 time units. The optimal schedule uses $k = 2$ maintenance periods and consists of a group with job $j = 4$ and a group with the other three jobs.

Dealing with problems $1/p(j,r), \mathcal{M}/\gamma$ and $\gamma \in \{\sum C_j, TADC\}$ requires more care. Remember that the evaluation of entry $f(j,r)$ also depends on the length l, measured in numbers of jobs, *i.e.* to assign job j in rank r we have to know the corresponding group i and its length. For $\sum C_j$ and maintenance MP we have for instance $f(j,r) = (l-r+1)p(j,r)$. One might try to introduce fictitious zero-length jobs ($p(j,r) = 0$ for all ranks r). But this would distort the problem since zero-length jobs would be located at the beginning of each group and would push the following (real) jobs up in their rank, thus changing the processing times.

It is preferable to us the assignment model with n rows (jobs). The n columns are listed as follows: Having k maintenance periods, there should be $n - (l-1)k$ groups, $l = \lceil \frac{n}{k} \rceil$, with l jobs and $lk - n$ groups with $l - 1$ jobs in any balanced schedule. List the n columns group by group with increasing ranks. Notice that now all entries $f(j,r)$ in each group are well defined since the group lengths, *i.e.* l or $l-1$ in this case, are known. Then we let the algorithm for this assignment problem place optimally the jobs in the groups and in their rank. We get again alternate optimal solutions by exchanging the processing terms in the same rank between two groups, provided the groups have the same lengths l or $l-1$. Notice that we only have a given sequence β_i of the constant parts of the maintenance periods. The sequence, consisting of the processing in group i and the variable part of the following maintenance period i can be changed arbitrarily to obtain alternate optimal solutions.

We have to solve an assignment problem for each k, $1 \leq k \leq n$, to get an overall complexity $O(n^4)$.

Proposition 4 *An optimal balanced schedule for the problems*

$$1/p(j,r), \mathcal{M}/\gamma \textit{ with } \gamma \in \{\sum C_j, TADC\}$$

can be found in $O(n^4)$ time.

Balanced schedules are important in connection with preventive maintenance, where one would like to do maintenance work on a regular basis after a given number l of parts produced. Suppose there are n parts and l is given. We like to have group sizes as close as possible to l but not exceeding l. Thus we can use k maintenance periods, where k satisfies $l = \lceil \frac{n}{k} \rceil$. Then the previous method finds in $O(n^3)$ time the best balanced schedule, in which at most l parts are processed between consecutive maintenance periods (in fact there are l or $l-1$ parts processed between any two periods, depending on the number n).

3.2 Unbalanced optimal schedules

Optimal schedules for the criteria $\sum C_j$ and $TADC$ are not necessarily balanced. Thus to determine an optimal schedule, our basic hypothesis is no longer satisfied: we can only set up the entries $f(j,r)$ in a given group if we know in advance the number of jobs allocated to this group. This also means the following: Suppose we have k maintenance periods and every maintenance crew has specified the amount of processing work, after which they want to perform their maintenance. Here the processing work is measured in terms of the number of parts to be produced since the previous maintenance. Then the number n of jobs is partitioned into k groups with $l_1, \ldots, l_k$ jobs, *i.e.* $n = \sum_{i=1}^{k} l_i$. Now we can determine in $O(n^3)$ time the best schedule with k maintenance periods, each of them with the specified processing work (l_i parts for group i) since the previous maintenance, using the same scheme as in the previous section. Let us denote these scheduling problems by $1/p(j,r), \mathcal{M}, (k, l_i)/\gamma$.

Proposition 5 *An optimal schedule for the problems*

$$1/p(j,r),\mathcal{M},(k,l_i)/\gamma \text{ with } \gamma \in \{\sum C_j, TADC\}$$

can be found in $O(n^3)$ time.

This result is generalizing and simplifying Theorem 1 in [14]. There a lengthy proof is given for the special case $p_j r^{a_j}$, $MP(\tau)$ and $\sum C_j$. Then all possible workloads $l_1,\dots,l_k$ ($n = \sum_{i=1}^k l_i$) are taken to get the best schedule in $O(n^{k_0+3})$ time, where k_0 is an upper bound on the number of maintenance periods.

3.3 Group dependent schedules

We describe a further extension of the models that applies only to the criterion C_{max} and has been proposed in [11]. The maintenance activity does not fully restore the machine to its initial ideal state. We obtain processing times that are *group dependent*.

The following model is given in [11]. The maintenance period i is preceded by the group of jobs i ($i = 1,\dots,k$) if there are k maintenance periods. The processing times in group i are of the form $p_j g_j^{[i]}(r)$ and the factors ${g_j}^{[i]}(r)$ are deteriorating with increasing rank r for each job j and in each group i. It is also assumed that factors are deteriorating with increasing groups: $g_j^{[1]}(r) \le g_j^{[2]}(r) \le \dots \le g_j^{[k]}(r)$. One might have any of the previous maintenance systems $\mathcal{M}$. But also a more general time dependent maintenance system is presented, which we may call $MP(\tau,i)$, whose durations are given in the form $t_i = \alpha_i\tau_i + \beta_i$ with $\alpha_1 \le \alpha_2 \le \dots \le \alpha_k$. Notice that reading of input data requires already $O(n^3)$ time.

We shall slightly generalize and simplify this model, but follow in essence the approach of [11]. Define group dependent processing times $p(j,r;i)$, deteriorating with the rank r in each group i and for each job j. There is no imposed correlation between processing times in different groups. One just has a different processing environment for each group, due to varying quality of maintenance and eventually different production operators in manual processing systems from one group to another. There may be different maintenance teams with different skills and experience. In model MP their maintenance duration is constant ($t_i = \beta_i$), restoring fully or partially the ideal state of the machine. For model $MP(\tau,i)$ and more general for a weighted maintenance model $MP(w,i)$ we may argue as follows: the maintenance crew i can evaluate a weight factor $w(j,r;i)$, deteriorating with r for each job j, which requires the maintenance duration $w(j,r;i)p(j,r;i)$ in in period i, in addition to some constant duration β_i. Then, as previously done, we replace the processing times $p(j, r; i)$ with $(1 + w(j, r; i))p(j, r; i)$ in group i and transform model $MP(w, i)$ to model MP. Notice that we get the model of [11] by setting $w(j, r; i) = \alpha_i$. The optimal schedules are no longer balanced, even for C_{max}. Therefore, we have to permit that the number of jobs in any group i may be as large as the number n of jobs.

Observation 4 *([4], [9], see also [12] P. 290) Let a complete bipartite graph $G_{n,m}$, $n \leq m$, with weights $c_{j,i}$ on the edges be given. Then a minimum-weight matching in $G_{n,m}$ of size n can be determined in $O(n^3 + nm)$ time.*

([3]) As alternative one can describe the same problem as rectangular assignment problem of size nxm, where the n row sums are equal to 1 and the m column sums are less or equal to 1. An algorithm with the weaker complexity $O(n^2m)$ is given.

For k maintenance periods we set up an nx(nk) table, n free places in each of the k groups, and let the matching algorithm fill optimally the positions with complexity $O(n^3 + n^2k) = O(n^3)$ for all k. This method applies only to C_{max} since the number of jobs allocated to each group is not known in advance. The overall complexity is again $nO(n^3) = O(n^4)$.

Proposition 6 *An optimal schedule for the problems*

$$1/p(j, r; i), \mathcal{M}/C_{max} \text{ with } \mathcal{M} \in \{MP, MP(\tau, i), MP(w, i)\}$$

can be found in $O(n^4)$ time.

The running time in [11] for the functions $p_j g_j^{[i]}(r)$ is given as $O(n^5)$ since the weaker algorithm of [3] is used, which has the complexity $O(n^3k)$ for k maintenance periods.

3.4 Final remarks

The group dependent weights $w(j, r; i)$ yield the most flexibility. They may be used for group dependent processing times $p(j, r; i)$ but also in connection with processing times $p(j, r)$. Suitable choices for the weights allow within the same production process to modify the maintenance system $\mathcal{M}$ (having periods with constant maintenance times and others with time dependent maintenance). We have imposed a closed maintenance system (a period at the end). Now one can change easily to an open-end maintenance (no maintenance at the end), simply by setting in the final period k the weights $w(j, r; k) = 0$ and $t_k = 0$.

Let us return to balanced schedules, processing times $p(j,r)$ and general weights $w(j,r;i)$. Let us first consider an optimal balanced schedule for the makespan C_{max}. For each number k of maintenance periods we can add zero-length jobs so that we get k groups, each with exactly $l = \lceil \frac{n}{k} \rceil$ jobs. The corresponding assignment problem gives the optimal balanced solution for all choices of weights $w(j,r;i)$. For instance with the scheme above ($w(j,r;k) = 0$ and $t_k = 0$) one eliminates the maintenance period at the end.

However, for open-end maintenance systems an optimal balanced schedule for C_{max} is not necessarily an overall optimal solution. Consider the following example: Take $n = 4$ jobs. Three jobs are identical: $p(j,r) = 1, 2, 2, 5$ ($j = 1, 2, 3$) and $p(4,r) = 10, 20, 20, 20$. The maintenance periods take $t_i = \alpha\tau_i + \beta = 2\tau_i$ time units for $i \neq k$ and $t_k = 0$. The optimal balanced schedule uses $k = 3$ maintenance periods and is as follows: (job 1; 2 units maintenance; job 2; 2 units maintenance; job 4, job 3; no maintenance) and $C_{max} = 18$. The optimal solution uses $k = 2$ maintenance periods: (job 1; 2 units maintenance; job 4, job 2, job 3; no maintenance) and $C_{max} = 17$.

For the other two criteria, $\sum C_j$ and $TADC$, balanced schedules would have in addition to general weights also changing group lengths, *i.e.* $n-(l-1)k$ groups with l jobs and $lk - n$ groups with $l - 1$ jobs. We may specify all weights and group lengths to get the best balanced schedule under these assumptions. Suppose that we want to eliminate the maintenance period at the end. Then we set $w(j,r;i) = w(j,r)$ for all $i \neq k$, $w(j,r;k) = 0$ and $t_k = 0$. To get now an optimal balanced schedule, we have to solve for each k two assignment problems, one with the final group length l and the second with length $l - 1$ and choose the better schedule.

References

[1] J. Bang-Jensen, G. Gutin. Digraphs, Theory, Algorithms and Applications. Springer, 2002.

[2] D. Biskup. A state-of-the-art review on scheduling with learning effects. European Journal of Operational Research. 1999, Vol. 188, P. 315-29.

[3] R. Burkard, R.M. Dell'Amico, S. Martello. Assignment problems. Philadelphia, SIAM, 2009.

[4] E.A. Dinic. On solution of two assignment problems. In: *Studies in Discrete Optimization*, A.A. Fridman ed., "Nauka", Moscow. 1976, P. 333-48 (in Russian).

[5] E.A. Dinic, M.A. Kronrod. An algorithm for the solution of the assignment problem. Soviet Math. Dokl. 1969, Vol. 10, 6, P. 1324-26.

[6] V.S. Gordon, V.A. Strusevich. Single machine scheduling and due date assignment with positionally dependent processing times. European Journal of Operational Research. 2009, Vol. 198, P. 57-62.

[7] G.H. Hardy, J.E. Littlewood, G. Polya. Inequalities. London, Cambridge University Press, 1934.

[8] W.H. Kuo, D.L. Yang. Minimizing the makespan in a single-machine scheduling problem with the cyclic process of an aging effect. Journal of the Operational Research Society. 2008, Vol. 59, P. 416-20.

[9] N. Milo, S. Zakov, E. Katzenelson, E Bachmat, Y. Dinitz, M. Ziv-Ukelson. Unrooted unordered homeomorphic subtree alignment of RNA trees. Algorithms for Molecular Biology. 2013, Vol. 8, 13.

[10] J. Moncel, G. Finke, V. Jost. A note on single-machine scheduling problems with position-dependent processing times. 13th Int. Conf. on Project Management and Scheduling - PMS 2012. Leuven, Belgium.

[11] K. Rustogi, V. Strusevich. Single machine scheduling with general positional deterioration and rate-modifying maintenance. OMEGA. 2012, Vol. 40, P. 791-804.

[12] A. Schrijver. Combinatorial Optimization, Polyhedra and Efficiency. Vol. A, Springer, 2003.

[13] S.-J. Yang, D.-L Yang. Minimizing the makespan on single-machine scheduling with aging effect and variable maintenance activities. OMEGA. 2010, Vol. 38, 6, P. 528-33.

[14] S.-J. Yang, D.-L Yang. Minimizing the total completion time in single-machine scheduling with aging/deteriorating effects and deteriorating maintenance activities. Computers and Mathematics with Applications. 2010, Vol. 60, P. 2161-69.

Les cahiers Leibniz ont pour vocation la diffusion des rapports de recherche, des séminaires ou des projets de publication sur des problèmes liés au mathématiques discrètes.

Responsables de la publication : Nadia Brauner et András Sebő
ISSN : 1298-020X