SEQUENCE-DEPENDENT SETUP AND CLEAN-UP TIMES IN A TWO-MACHINE JOB-SHOP WITH MINIMIZING MAKESPAN

Yuri N. Sotskov [*] **Frank Werner** [**]

[*] *United Institute of Informatics Problems, National Academy of Sciences of Belarus, Surganova Str. 6, Minsk, Belarus, e-mail: sotskov@newman.bas-net.by*
[**] *Faculty of Mathematics, Otto-von-Guericke University, Postfach 4120, D-39016, Magdeburg, Germany, e-mail: frank.werner@mathematik.uni-magdeburg.de*

Abstract: This article addresses the job-shop scheduling problem of minimizing the length of a schedule (makespan) for processing n jobs by one or two machines with sequence-dependent setup times and clean-up times. The processing of each job includes at most two operations that have to be non-preemptive. Machine routes may differ from job to job. If all setup and clean-up times are equal to zero, this problem is polynomially solvable via Jackson's pair of job permutations, otherwise it is NP-hard even if each of n jobs consists of one operation on the same machine. We present sufficient conditions when Jackson's pair of job permutations may be used for solving the two-machine job-shop scheduling problem with sequence-dependent setup times and clean-up times. For the general case of the latter problem, the results obtained provide polynomial lower and upper bounds for the objective function which may be used in an implicit enumeration technique, e.g., in a branch-and-bound algorithm. *Copyright © IFAC 2006*

Keywords: Scheduling algorithms, Optimization problems, Operation research

1. INTRODUCTION

The majority of scheduling research assumes the *setup time* as negligible or as a part of the job processing time. This assumption adversely affects the solution quality for many applications which require an explicit treatment of setup times. Practical situations in which setup times must be considered separately from processing times arise in chemical, pharmaceutical, food, printing, metal processing and semiconductor industries. These applications have motivated an increasing interest to include separate setups in the scheduling environment, see e.g. in (Allahverdi *et al.*, 1999). In (Khurana and Bagga, 1984; Yoshida and Hitomi, 1979), the authors addressed the two-machine flow-shop problem of minimizing C_{max}, the length of a schedule (makespan), by considering setup times separately. In (Allahverdi, 2000; Bagga and Khurana, 1986), the two-machine separate setup times problem of minimizing mean job completion time $\sum C_i$ has been addressed. A two-machine open-shop problem with inter-stage transportation times has been considered in (Strusevich, 1999). Allahverdi (2000) adressed the two-machine flow-shop problem to minimize C_{max} or $\sum C_i$ when setup times are relaxed to be distribution-free random variables with lower and upper bounds being given before scheduling. In (Sotskov *et al.*, 1999), insertion techniques were used for a job-shop problem with sequence-independent setup times.

As follows from (Allahverdi *et al.*, 2003) and the other published reviews of scheduling research with separate setup times, only shop-scheduling problems involving *sequence-independent* setup times have been treated in the OR literature so far. While the assumption that setup times are sequence-independent simplifies the analysis of a shop-scheduling problem and reflects certain applications, it negatively affects the solution quality for many other applications which require a treatment of *sequence-dependent* setup times. In this article, we consider the two-machine job-shop scheduling problem of minimizing the length C_{max} of a schedule including sequence-dependent setup times and clean-up times. Note that the clean-time is often referred to as removal time in the OR literature.

Assume that a set of jobs $J = \{1, 2, \dots, n\}$ has to be processed in a job-shop with two machines $M = \{1, 2\}$ provided that each machine $m \in M$ processes any job $j \in J$ at most once. Subset J_{12} of set J is the set of all jobs with machine route $(1, 2)$. $J_{21} \subseteq J$ is the set of all jobs with opposite machine route $(2, 1)$, and $J_m \subseteq J$ is the set of all jobs which have to be processed only on one machine $m \in M$. Thus, we have

$$J = J_{12} \cup J_1 \cup J_2 \cup J_{21}.$$

The cardinality of set J_k is denoted as $n_k = |J_k|$, where $k \in \{1, 2, 12, 21\}$. As usual, O_{jm} denotes the operation of job $j \in J$ on machine $m \in M$. The processing time p_{jm} of operation O_{jm} is known before scheduling. All n jobs are available for processing from time $t = 0$. Operation preemptions are forbidden.

In practice, machines often have to be reconfigured before starting a job and cleaned after completing the last job. These processes are called *setup* and *clean-up*. We assume that the given setup time of a machine depends on the job just completed and the job to be started, i.e., the given setup times are sequence-dependent. If job $i \in J$ is directly followed by job $k \in J$ on machine $m \in M$, then the setup time is equal to a non-negative real number s^m_{ik}.

Similarly, the notation s^m_{0k} is used for the setup time needed on machine $m \in M$ before starting job k, if job k is the first job processed on machine m. On the other hand, s^m_{i0} denotes the clean-up time after job i, if job i is the last job processed on machine $m \in M$.

Setup and clean-up times for machine 1 are given by a real non-negative square matrix $S^1 = ||s^1_{ij}||$ of order $r_1 \times r_1$, where $r_1 = n - n_2 + 1$. Hereafter, in contrast to usual matrix notations when the subindex i (subindex j) of the element s^1_{ij} of matrix S^1 denotes the row index (column index, respectively), we define that the first subindex i in s^1_{ij} denotes job $i \in J \setminus J_2$ and the second subindex j in s^1_{ij} denotes job $j \in J \setminus J_2$: If job i is directly followed by job $j \neq i$ in the processing sequence of r_1 jobs on machine 1, then the corresponding setup time on machine 1 is just equal to s^1_{ij}.

As usual, it is assumed that columns (rows) in matrix S^1 are ordered with respect to an increasing second subindex (first subindex) of their elements s^1_{ij}. Thus, each element s^1_{0i} of the first row in matrix S^1 defines the setup time for job $i \in J \setminus J_2$ on machine 1, if job i is the first job processed on machine 1. On the other hand, each element s^1_{j0} of the first column in matrix S^1 defines the clean-up time for job $j \in J \setminus J_2$, if job j is the last job processed on machine 1.

It is clear that the diagonal elements (i.e., those with equal first and second subindices) in matrix S^1 are not used.

Similarly, setup and clean-up times for machine 2 are given by a real non-negative square matrix $S^2 = ||s^2_{ij}||$ of order $r_2 \times r_2$, where $r_2 = n - n_1 + 1$.

Since the minimization of the schedule length is a *regular* criterion, one can consider only *semiactive schedules*, see e.g. (Tanaev *et al.*, 1994) for definitions. Each semiactive schedule is uniquely defined by a permutation of the jobs on machine 1 and by one on machine 2. Thus, the problem under consideration is to find a permutation

$$\pi' = (i'_1, i'_2, \dots, i'_{r_1})$$

of the jobs $i'_k \in J_{12} \cup J_1 \cup J_{21}$ on machine 1 and a permutation

$$\pi'' = (i''_1, i''_2, \dots, i''_{r_2})$$

of the jobs $i''_k \in J_{12} \cup J_2 \cup J_{21}$ on machine 2 that minimize the objective function

$$C_{max}(\pi', \pi'') =$$

$$\max\{C_{i'_{r_1}}(\pi', \pi'') + s^1_{i'_{r_1}0},\ C_{i''_{r_2}}(\pi', \pi'') + s^2_{i''_{r_2}0}\}, \quad (1)$$

where $C_i(\pi', \pi'')$ denotes the completion time of job $i \in J$ in the semiactive schedule defined by the pair of permutations (π', π''). Objective function (1) defines the schedule length including clean-up times after the last jobs. This problem is denoted as $J2|s_{jk}|C_{max}$.

2. MODIFICATION OF SETUP, CLEAN-UP (REMOVAL) AND PROCESSING TIMES

The value of objective function (1) depends on two essentially different parts of the numerical input data. The first part includes the processing times p_{ij} of jobs $i \in J$ on machines $j \in M$, while the second part includes the setup and clean-up times given by the square matrices S^1 and S^2. Generally speaking, the former part is easier to treat optimally than the latter part. Indeed, if all setup times and clean-up times are equal to zero, then problem $J2|s_{jk}|C_{max}$ turns into the classical job-shop problem $J2||C_{max}$ which is polynomially solvable by Jackson's pair of job permutations, see (Jackson, 1956), otherwise problem $J2|s_{jk}|C_{max}$ is NP-hard even if each of the n jobs consists of one operation on the same machine, e.g., if $n = n_1$. (The latter problem turns into the NP-hard traveling salesman problem.)

If there exist non-zero setup or clean-up times, then the schedule length $C_{max}(\pi', \pi'')$ essentially depends on the choice of $r_1 + r_2$ setup and clean-up times (from the set of all $r_1^2 + r_2^2$ possible setup and clean-up times given by matrices S^1 and S^2) which have to be involved into the schedule.

In this section, we show how it is possible to transfer at least a part of the "hard" numerical input data to the "easy" numerical input data.

Let job i belong to set $J_1 \cup J_{12}$. We calculate the non-negative value

$$s^1(\rightarrow i) = \min\{s^1_{ki} \mid i \neq k \in \{0\} \cup J \setminus J_2\}. \quad (2)$$

Since each setup time before processing operation O_{i1} includes a part equal to $s^1(\rightarrow i)$, we can add this value $s^1(\rightarrow i)$ to processing time p_{i1} of operation O_{i1} provided that the same value $s^1(\rightarrow i)$ will be subtracted from each setup time s^1_{ki} with $i \neq k \in \{0\} \cup J \setminus J_2$. Thus for each job $i \in J_1 \cup J_{12}$, we obtain the following *modified* processing time:

$$p'_{i1} = s^1(\rightarrow i) + p_{i1} \quad (3)$$

and the following *modified* setup times:

$$s^{(1)}_{ki} = s^1_{ki} - s^1(\rightarrow i), \quad i \neq k \in \{0\} \cup J \setminus J_2. \quad (4)$$

Due to equalities (2) and (4), we obtain $s^{(1)}_{ki} \geq 0$ for each $i \in J_1 \cup J_{12}$ and $k \in \{0\} \cup J \setminus J_2$ with $k \neq i$.

Next, we show that the *original* instance of problem $J2|s_{jk}|C_{max}$ and the *modified* instance that differs from the *original* instance only by the setup and processing times of jobs $i \in J_1 \cup J_{12}$ modified due to equalities (3) and (4) are *equivalent* in the following sense.

Two instances of a scheduling problem are equivalent if there exists a one-to-one correspondence between their semiactive schedules such that the corresponding two schedules have the same value of the objective function.

Indeed, the desired correspondence of semiactive schedules is defined by the same pair (π', π'') of permutation

$$\pi' = (i'_1, i'_2, \ldots, i'_{r_1})$$

of jobs $i'_k \in J_{12} \cup J_1 \cup J_{21}$ on machine 1 and permutation

$$\pi'' = (i''_1, i''_2, \ldots, i''_{r_1})$$

of jobs $i''_k \in J_{12} \cup J_2 \cup J_{21}$ on machine 2. It is easy to convince that for both instances of problem $J2|s_{jk}|C_{max}$, machines 1 and 2 are occupied (either by processing jobs or by setups) during the same time intervals since in each semiactive schedule constructed for the *modified* instance each non-negative value $s^1(\rightarrow i)$ is added exactly once to processing time p_{i1} and subtracted exactly once from the setup time which is involved in the schedule. Moreover, the processing time p_{i1} of each job $i \in J_{12}$ is increased only "from the left-hand side" by the value $s^1(\rightarrow i)$ of the setup time defined by equality (2). Hence, the processing of job $i \in J_{12}$ on machine 2 may be started just from the same time as in the corresponding semiactive schedule constructed for the *original* instance of problem $J2|s_{jk}|C_{max}$.

Due to machine symmetry, one can also obtain an equivalent *modified* instance of problem $J2|s_{jk}|C_{max}$ via modifying the setup and processing times of jobs $i \in J_2 \cup J_{21}$ on machine 2:

$$p'_{i2} = s^2(\rightarrow i) + p_{i2}, \quad (5)$$

$$s^{(2)}_{ki} = s^2_{ki} - s^2(\rightarrow i), \quad i \neq k \in \{0\} \cup J \setminus J_1. \quad (6)$$

The above value $s^2(\rightarrow i)$ is defined as follows:

$$s^2(\rightarrow i) = \min\{s^2_{ki} \mid i \neq k \in \{0\} \cup J \setminus J_1\}. \quad (7)$$

Similarly, one can increase the processing times of the jobs in set J_{21} on machine 1 "from the right-hand side" due to the decrease of the corresponding setup and clean-up times as follows.

Let job j belong to set $J_1 \cup J_{21}$. We calculate the non-negative value

$$s^1(j \rightarrow) = \min\{s^1_{jk} \mid j \neq k \in \{0\} \cup J \setminus J_2\}. \quad (8)$$

Since the clean-up time and each setup time before operation O_{j1} includes a part equal to $s^1(j \rightarrow)$, we can add this value $s^1(j \rightarrow)$ to processing time p_{j1} of operation O_{j1} provided that the same value $s^1(j \rightarrow)$ will be subtracted from the clean-up time s^1_{j0} and from each setup time s^1_{jk} with $j \neq k \in J \setminus J_2$. Thus for each job $j \in J_1 \cup J_{21}$, we obtain the *modified* processing time

$$p'_{j1} = p_{j1} + s^1(j \rightarrow), \quad (9)$$

the *modified* setup times

$$s^{(1)}_{jk} = s^1_{jk} - s^1(j \rightarrow), \quad j \neq k \in J \setminus J_2, \quad (10)$$

and the *modified* clean-up time

$$s_{j0}^{(1)} = s_{j0}^1 - s^1(j \rightarrow). \tag{11}$$

Note that due to equality (9), the processing time p_{j1} of job $j \in J_{21}$ is increased only "from the right-hand side" by the value $s^1(j \rightarrow)$ defined by (8). Due to this and equalities (2)-(3), the non-negative common part of each setup time may be added to the modified processing time exactly once. Note that the processing times of jobs $i = j \in J_1$ may be modified both "from the left-hand side" due to equalities (2)-(3) used for job $i \in J_1$ and "from the right-hand side" due to equalities (8)-(9) used for job $j \in J_1$.

Similarly (due to machine symmetry), one can modify setup, clean-up, and processing times of jobs $i \in J_2 \cup J_{12}$ on machine 2 using formulas (12)-(14):

$$p'_{i2} = p_{i2} + s^2(i \rightarrow), \tag{12}$$

$$s_{ik}^{(2)} = s_{ik}^2 - s^2(i \rightarrow), \ \ i \neq k \in \{0\} \cup J \setminus J_2, \tag{13}$$

$$s_{j0}^{(2)} = s_{j0}^2 - s^2(j \rightarrow), \tag{14}$$

where value $s^2(i \rightarrow)$ is defined as follows:

$$s^2(i \rightarrow) = \min\{s_{ik}^2 \mid i \neq k \in \{0\} \cup J \setminus J_2\}. \tag{15}$$

In order to transfer further "hard" numerical input data to the "easy" numerical input data, we can introduce a *dummy* job 0 (and a *dummy* job $n+1$, respectively) before starting the first (and after completing the last) job on each of the two machines. The processing times p_{0m} and the modified setup times $s_{0j}^{(m)}$ are defined as follows:

$$p_{0m} = s^m(0), \tag{16}$$

$$s_{0j}^{(m)} = s_{0j}^m - s^m(0), \ \ j \in J \setminus J_{3-m}, \tag{17}$$

provided that

$$s^m(0) = \min\{s_{0j}^m \ \mid \ j \in J \setminus J_{3-m}\}. \tag{18}$$

Similarly, the processing times $p_{n+1,m}$ and the modified clean-up times $s_{j0}^{(m)}$ are defined as follows:

$$p_{n+1,m} = s^m(n+1), \tag{19}$$

$$s_{j0}^{(m)} = s_{j0}^m - s^m(n+1), \ \ j \in J \setminus J_{3-m}, \tag{20}$$

provided that

$$s^m(n+1) = \min\{s_{j0}^m \ \mid \ j \in J \setminus J_{3-m}\}. \tag{21}$$

Thus, the following claim has been proven.

Theorem 1. An instance of problem $J2|s_{jk}|C_{max}$ is equivalent to the modified instance that differs from the original one only by setup, clean-up, and processing times of jobs $J \cup \{0, n+1\}$ modified due to formulas (2)-(21).

In order to obtain the simplest *modified* instance (using formulas (2)-(21)), which is equivalent to the *original* instance of problem $J2|s_{jk}|C_{max}$, it is necessary to decrease the elements of the square matrices S^1 and S^2 as much as possible. Therefore, the simplest equivalent *modified* instance is obtained when no further modification of the matrices based on formulas (2)-(21) is possible.

Let matrix $S^{(1)}$ and matrix $S^{(2)}$ denote such a *minimal* matrix (its elements have minimal feasible values) obtained from matrix S^1 and matrix S^2, respectively, by means of formulas (2)-(21).

Note that the *minimal* matrices $S^{(1)}$ and $S^{(2)}$ are uniquely defined, while there may exist several *modified* instances of the *original* instance of problem $J2|s_{jk}|C_{max}$ (because of different orders that may be used for the modification of rows and columns of the given matrices S^1 and S^2). We say:

An instance of problem $J2||C_{max}$ corresponds to an instance of problem $J2|s_{jk}|C_{max}$ (and vice versa), if all the parameters and conditions are the same for both instances except non-zero setup and clean-up times given for the instance of problem $J2|s_{jk}|C_{max}$.

Machine $m \in M$ is called the *main machine* for the semiactive schedule defined by the pair (π', π'') of permutations, if the following equality holds:

$$C_{max}(\pi', \pi'') = C_j(\pi', \pi'') + s_{j0}^m,$$

where $j = i'_{r_1}$ if $m = 1$, and $j = i''_{r_2}$ if $m = 2$.

Let $c_j^m(\pi', \pi'')$ denote the completion time of operation O_{jm} in the semiactive schedule defined by (π', π''). Now, using the arguments given in (Braun *et al.*, 2006) and the above Theorem 1, we can prove the following sufficient conditions for optimality of Jackson's pair of job permutations for problem $J2|s_{jk}|C_{max}$.

Theorem 2. Jackson's pair (π', π'') of job permutations constructed for the instance of problem $J2||C_{max}$ with job processing times obtained by means of formulas (2), (3), (5), (7), (8), (9), (12), (15) (16), (18), (19), (21) remains optimal for the corresponding instance of problem $J2|s_{jk}|C_{max}$, if the main machine $m \in M$ for the semiactive schedule defined by (π', π'') has no idle times and has only zero modified setup and clean-up times.

Theorem 3. Jackson's pair (π', π'') of job permutations constructed for the instance of problem $J2||C_{max}$ with job processing times obtained by means of formulas (2), (3), (5), (7), (8), (9), (12), (15), (16), (18), (19), (21) remains optimal for the corresponding instance of problem $J2|s_{jk}|C_{max}$, if there exists a time $t = c_j^m(\pi', \pi'')$ such that the following two conditions hold:
(i) For the semiactive schedule defined by (π', π''), machine $m \in M$ has no idle times and has only zero modified setup times in the segment $[0,\ t]$;
(ii) In the segment $[t,\ C_j(\pi', \pi'') + s_{j0}^{3-m}]$ with $j = i'_{r_1}$ for $m = 2$ and $j = i''_{r_2}$ for $m = 1$, the main machine $(3 - m) \in M$ for the semiactive schedule defined by (π', π'') has no idle times and has only zero modified setup and clean-up times.

Theorem 1 implies the following sufficient condition for optimality of Jackson's pair (π', π'') of permutations for problem $J2|s_{jk}|C_{max}$ with any given job processing times: *The minimal matrices* $S^{(1)} = ||s_{ij}^{(1)}||$ *and* $S^{(2)} = ||s_{ij}^{(1)}||$ *have only zero elements*: $s_{ij}^{(1)} = 0 = s_{ij}^{(2)},\ i \neq j$.

If it is a priory clear which machine $m \in M$ has to be the main machine in the semiactive schedule defined by (π', π'') without idle times on machine m, then the above sufficient condition is reduced to the following: *The minimal matrix* $S^{(m)}$ *has only zero elements.*

3. WORST CASE ANALYSIS

Using the results presented in Section 2, we propose the following algorithm for solving problem $J2|s_{jk}|C_{max}$ exactly or for finding an approximate solution to the problem $J2|s_{jk}|C_{max}$.

1. Construct a *modified* instance that is equivalent (due to Theorem 1) to the *original* instance of problem $J2|s_{jk}|C_{max}$.

2. Find Jackson's pair (π', π'') of job permutations constructed for problem $J2||C_{max}$ corresponding to the *modified* instance of problem $J2|s_{jk}|C_{max}$.

3. Test the sufficient conditions for optimality of (π', π'') for the *modified* instance of problem $J2|s_{jk}|C_{max}$ given by Theorem 2, Theorem 3, and so on.

4. If at least one of the sufficient conditions holds, the *original* instance of problem $J2|s_{jk}|C_{max}$ is solved by the pair (π', π'') of job permutations. Otherwise, the semiactive schedules constructed for the corresponding instance of problem $J2||C_{max}$ (those constructed for the corresponding instance of problem $J2|s_{jk}|C_{max}$) polynomially provide lower bounds (upper bounds, respectively) for the objective function (1).

The former lower bound for the objective function (1) may be used in an implicit enumeration technique, e.g., in a branch-and-bound algorithm developed for problem $J2|s_{jk}|C_{max}$. A worst case analysis of the solution obtained using the algorithm consisting of steps 1 - 4 shows the following results.

Let C^*_{max} denote the optimal value of the objective function (1), and $C_{max}(\pi', \pi'')$ denote the value of the objective function (1) obtained using the algorithm consisting of steps 1 - 4. We denote

$n_{min} = \min\{\min\{|J \setminus J_1|, |J \setminus J_2|\}, \min\{|J_{12}| + 1, |J_{21}| + 1\}\}$,

$n_{max} = \max\{\max\{|J \setminus J_1|, |J \setminus J_2|\}, \max\{|J_{12}| + 1, |J_{21}| + 1\}\}$,

$s_{min} = \min\{s_{ij}^{(m)} \mid m \in M, i \in J,\ i \neq j \in J\}$,

$s_{max} = \max\{s_{ij}^{(m)} \mid m \in M, i \in J,\ i \neq j \in J\}$.

The above value n_{min} (n_{max}, respectively) defines the minimal (maximal) cardinality of the critical set of operations which defines the objective value $C_{max}(\pi', \pi'')$.

If

$$s_{ij}^{(m)} \leq p_j^{(m)},\ i \in J,\ i \neq j \in J, \qquad (22)$$

then

$$C_{max}(\pi', \pi'') \leq 2C^*_{max} - n_{min}s_{min}.$$

If

$$p_j^{(m)} \leq s_{ij}^{(m)} \leq 2p_j^{(m)},\ i \in J,\ i \neq j \in J, \qquad (23)$$

then

$$C_{max}(\pi', \pi'') \leq 3/2C^*_{max}.$$

In the case when both conditions (22) and (23) do not hold, we obtain the following upper bound:

$$C_{max}(\pi', \pi'') \leq C^*_{max} + n_{max}(s_{max} - s_{min}).$$

In the latter case, the heuristic rule based on setup and clean-up times may be more effective than that based on the modified processing times considered in Section 2.

4. CONCLUSION

In most of the shop-scheduling models considered in the OR literature, it is assumed that an individual processing time incorporates all other time parameters (lags) attached to a job or to an operation. In practice, however, such parameters often have to be considered separately from the actual processing times. For example, if for an operation some pre-processing and post-processing are required, then it is necessary to use a scheduling model with setup and clean-up (removal) times separated. Moreover, setup times are often sequence-dependent.

In Sections 2 and 3, we derived sufficient conditions when Jackson's pair of job permutations may be used for solving the two-machine job-shop scheduling problem with sequence-dependent setup times and removal times. In a forthcoming paper, we will present computational results for the cases when our approach is better than the existing ones.

This research was supported by INTAS (project 03-51-5501) and ISTC (project B-986). The authors would like to thank three anonymous referees for their suggestions on an early version of the paper.

REFERENCES

Allahverdi, A. (2000). Minimizing mean flowtime in a two-machine flowshop with sequence independent setup times. *Comp. Oper. Res.* **27**, 111–127.

Allahverdi, A., J.N.D. Gupta and T. Aldowaisan (1999). A review of scheduling research involving setup considerations. *OMEGA, The International Journal of Management Sciences* **27**, 219–239.

Allahverdi, A., T. Aldowaisan and Yu.N. Sotskov (2003). Two-machine flowshop scheduling problem to minimize makespan or total completion time with random and bounded setup times. *International Journal of Mathematics and Mathematical Sciences* **39**, 2475–2486.

Bagga, P.C. and K. Khurana (1986). Two-machine flowshop with separated sequence-independent setup times: Mean completion time criterion. *Indian Journal of Management and Systems* **2**, 47–57.

Braun, O., N.M. Leshchenko and Yu.N. Sotskov (2006). Optimality of jackson's permutations with respect to limited machine availability. *International Transactions in Operational Research* **13**, 59–74.

Jackson, J.R. (1956). An extension of johnson's results on job lot scheduling. *Naval Research Logistics Quarterly* **3**, 201–203.

Khurana, K. and P.C. Bagga (1984). Minimizing the makespan in a two-machine flowshop with time lags and setup conditions. *Zeitschrift Operations Research* **28**, 163–174.

Sotskov, Yu.N., T. Tautenhahn and F. Werner (1999). On the application of insertion techniques for job shop problems with setup times. *RAIRO Recherche Operationnelle* **33**, 209–245.

Strusevich, V.A. (1999). A heuristic for the two-machine open-shop scheduling problem with transportation times. *Discrete Applied Mathematics* **93**, 287–304.

Tanaev, V.S., Yu.N. Sotskov and V.A. Strusevich (1994). *Scheduling Theory: Multi-Stage Systems.* Kluwer Academic Publishers. The Netherlands, Dordrecht.

Yoshida, T. and K. Hitomi (1979). Optimal two-stage production scheduling with setup times separated. *AIIE Transactions* **11**, 261–263.