Control and Cybernetics

vol. **38** (2009) No. 3

Heuristic algorithms for preemptive scheduling in a two-stage flowshop with unrelated parallel machines and 0-1 resource requirements[*]

by

Ewa Figielska

Warsaw School of Information Technology
ul. Newelska 6, 01-447 Warsaw, Poland
figielsk@wit.edu.pl

Abstract: The paper considers the problem of preemptive scheduling in a two-stage flowshop with parallel unrelated machines at the first stage and a single machine at the second stage. At the first stage, jobs use some additional renewable resources which are available in limited quantities. The resource requirements are of 0-1 type. The objective is minimization of the makespan. The problem is NP-hard. We develop heuristic algorithms which first solve the problem occurring at stage 1, and then find a final schedule in the flowshop. An extensive computational experiment shows that the proposed heuristic algorithms can be an efficient tool capable of finding good quality solutions.

Keywords: preemptive scheduling, flowshop, parallel unrelated machines, resource constraints, heuristics.

1. Introduction

In this paper, heuristic algorithms are proposed for solving the multiprocessor flowshop scheduling problem which can be described as follows: there are n preemptive jobs to be processed at two stages in the same technological order, first at stage 1 then at stage 2. At stage 1 there are m parallel unrelated machines, stage 2 has only one machine. A job, upon finishing its processing at stage 1, is ready to be processed at stage 2. At stage 1, a job can be processed on any of the parallel machines, and its processing times may be different on different machines. Processing times of job j are equal to p_{ij} (if executed on machine i) and s_j, respectively, at stage 1 and at stage 2. Processing of a job on a machine of stage 1 may be interrupted at any moment and resumed later on the same or another machine. A job during its processing at stage 1 does not use any additional resource or uses one unit of an additional resource. There are

[*]Submitted: October 2006; Accepted: December 2008

l types of additional resources. A resource of type r ($r = 1, \ldots, l$) is available in the amount limited to W_r units at a time. The total usage of resource r at any moment by the jobs that are simultaneously processed on parallel machines does not exceed W_r. The objective is to find a feasible schedule which minimizes the maximum job completion time in the two-stage flowshop, $C_{\max}$, referred to as makespan.

The considered problem is NP-hard in the strong sense since the problem of preemptive scheduling in the two-stage flowshop with two identical parallel machines at one stage and one machine at another is NP-hard in the strong sense (Hoogeveen et al., 1996).

The problem of scheduling in a flowshop with multiple machines, also called a hybrid flowshop, arises in real-life systems encountered in a variety of industries, e.g. in chemical, polymer, petrochemical industries (Salvador, 1973), as well as in computer systems and telecommunication networks (Brah, 1988). In the multiprocessor flowshop there are stages with parallel machines. At each stage with parallel machines jobs can be processed by any of these machines. Because jobs that are simultaneously processed on parallel machines may use the same resource, the problem of resource constrained scheduling arises when the amount of the available resource is limited. This takes place when, e.g., the number of workers attending the machines, or the number of tools that are used by simultaneously executed jobs, is limited. Resource requirements of 0-1 type can be met for example in computer systems in which one peripheral device (e.g. a printer) is additionally required to perform a job (Keller and Strusevich, 2002). The problems with preemptive jobs are common in mass production of a large number of products, which can be processed in parts or when an article is produced in a great amount, for example in the textile industry (Serafini, 1996) where processing of any job (the article to be woven) on one of the parallel machines (the looms) may be interrupted (preempted) and resumed on the same or another machine. The problem with parallel unrelated machines at the first stage and a single machine at the second stage may arise in a manufacturing environment in which products are initially processed on any of parallel machines and then each product must go through a final testing operation, which is to be carried out on a common testing machine.

During the last decade, the multiprocessor flowshops received considerable attention from researchers. Most literature in this area addresses the minimum makespan problems under the assumption that preemptions of jobs are not allowed and the parallel machines at each stage are identical (e.g. Gupta, 1988; Chen, 1995; Haouari and M'Hallah, 1997; Brah and Loo, 1999; Linn and Zhang, 1999). Only few papers concern the flowshop with parallel machines that are not identical (Suresh, 1997; Ruiz and Maroto, 2006).

In Janiak (1986, 1988a,b, 1989, 1991, 1998), Janiak and Portmann (1998), Grabowski and Janiak (1984) flowshop scheduling problems with resource constraints have been studied where processing times of jobs depend on the amount of resources used by these jobs. In the case when processing times are given a priori, in the one stage environment the problem of preemptive scheduling unrelated parallel machines under resource constraints has been widely investigated in the literature (e.g. Słowiński, 1980, 1981; de Werra, 1984; Figielska, 1999, 2005, 2006a). It is natural that such a problem may arise at some stages in multiprocessor flowshop environment.

In this paper, we consider the two-stage flowshop preemptive scheduling problem with unrelated parallel machines, renewable resource constraints and 0-1 resource requirements at the first stage and a single machine at the second stage. It is assumed that the jobs, while processing at the second stage do not require additional resources. For this problem, preliminary results have been presented in Figielska (2006b). The two-stage flowshop scheduling problem with arbitrary resource requirements has been studied in Figielska (2006c). For the considered problem, we propose effective heuristic algorithms which first solve the resource constrained scheduling problem at the first stage and then, taking into account the solution to this problem, construct the schedule at the second stage. The solution of the problem at the first stage has a form of a set of partial schedules which contain jobs processed in parallel under renewable resource constraints. Twelve sequencing procedures for ordering partial schedules are proposed to obtain possibly small makespan in the flowshop. In these procedures, the priority rules popular in the literature (the shortest processing time first (SPT), the longest processing time first (LPT), the Johnson algorithm, Johnson, 1954) are adapted for the use in the situation where each job is tied to a partial schedule. Proposed algorithms produce good quality results in a short computation time (few seconds) for problems with a large number of jobs.

The remainder of the paper is organized as follows. In the next section a heuristic algorithm is described in details. In Section 3 a lower bound is derived. Results of a computational experiment are reported and discussed in Section 4. Section 5 concludes the paper.

2. Heuristic algorithms

The heuristic algorithms proposed in this paper proceed in two steps, corresponding to two subproblems, which can be distinguished in the problem under consideration. The first of these subproblems, P1, is the resource constrained parallel machine scheduling problem occurring at the first stage of the two-stage flowshop. The second subproblem, P2, is a sequencing problem aiming at minimizing the makespan in the flowshop.

Problem P1, because of the structure of resource constraints (0-1 resource requirements of jobs), can be solved to optimality by the two-phase method proposed independently by Słowiński and Węglarz (1977) and Lawler and Labetoulle (1978) and extended for the cases with resource constraints by Słowiński (1981) and de Werra (1984). In this paper, to lessen the computational effort we use an approximate two-phase algorithm in which the first phase is the same as that in Słowiński (1981) (in this phase the makespan - the maximum job completion time - is minimized), and in the second phase the schedule is constructed by a constructive procedure so that the makespan at stage 1 is close to the optimal one.

A solution to problem P1 is composed of a number of partial schedules. (A partial schedule assigns some jobs, or parts of jobs, to machines for parallel processing during a certain period of time, so that resource constraints are fulfilled at every moment.) The makespan of this schedule does not depend on the ordering of the partial schedules, however, the times at which jobs finish their processing at stage 1 are different for different sequences of partial schedules. On the other hand, the makespan in the flowshop depends on job completion times at stage 1 (for each job the ready time at stage 2 is equal to its completion time at stage 1), so it depends on the ordering of partial schedules at stage 1.

Problem P2 is to find a sequence of the partial schedules which secures a flowshop schedule with short makespan. For solving this problem sequencing procedures are designed.

The proposed heuristic algorithms can be outlined as follows.

Step 1. Solving problem P1. The minimum makespan problem of unrelated parallel machines scheduling with additional resource constraints, which occurs at stage 1, is solved using an efficient approximate two-phase algorithm. The solution is composed of a number of partial schedules each satisfying renewable resource constraints.

Step 2. Solving problem P2. In order to obtain a flowshop schedule with small makespan a sequencing procedure is used for partial schedules.

2.1. Illustrative example

To illustrate the problem and the solution method we present the following example. Consider an instance of 10 jobs with processing times and resource requirements as shown in Fig. 1. The availability of the resource is one unit at a time, $W_1 = 1$. Stage 1 has two parallel unrelated machines, stage 2 has one machine. Fig. 2 presents two schedules for this instance. Each schedule in the two-stage flowshop can be treated as composed of a schedule of the first stage and a schedule of the second stage. The schedules of the first stage in Figs. 2a and 2b are composed of the same 8 partial schedules. Each partial schedule satisfies resource constraints at a time. For example, in the partial schedule of index 1, S_1, the total usage of the resource at any moment is equal to 1 (job 2 and 7 use 1 and 0 resource units, respectively). Similarly, all remaining partial schedules satisfy resource constraints at any moment.

In Fig. 2a the sequence of the partial schedules is (S_1, S_2,S_3, S_4, S_5, S_6, S_7, S_8). We can see that for this sequence of partial schedules, the first jobs finishing processing at stage 1 are jobs 2 and 7. After finishing its processing at stage 1, job 2 starts on the machine at stage 2. After completing job 2 at stage 2, job 7 starts at this stage (or first job 7 is processed, and then job 2). After completing jobs 2 and 7, the machine at stage 2 remains idle since it waits for finishing processing job 4 at stage 1. The order in which jobs finish processing at stage 1 is (2, 7, 4, 8, 6, 10, 1, 3, 5, 9). In Fig. 2a, we observe that the machine at stage 2 remains idle after the completion of jobs 7, 4 and 8 when it waits for finishing processing jobs 4, 8 and 6, respectively, at stage 1.

In Fig. 2b, the sequence of the partial schedules is (S_6, S_3, S_7, S_8, S_4, S_2, S_5, S_1). This sequence has been found by the SQ12 procedure (see Table 1). For this sequence of partial schedules, jobs finish their processing at stage 1 in the order (10, 3, 9, 1, 8, 4, 5, 6, 2, 7) at the times which are different than those in Fig. 2a, yielding a much shorter schedule. This is the optimal schedule: its makespan is equal to the lower bound on the optimal makespan.

job processing times at stage 1

	machine	
job	1	2
1	18	11
2	19	12
3	6	3
4	28	14
5	26	13
6	23	18
7	12	22
8	10	9
9	5	10
10	24	3

job processing times at stage 2

	machine
job	1
1	5
2	4
3	5
4	9
5	3
6	9
7	2
8	6
9	8
10	8

resource requirements at stage 1

job	
1	1
2	1
3	0
4	0
5	0
6	1
7	0
8	0
9	1
10	1

resource availability = 1

Figure 1. The data for an illustrative example

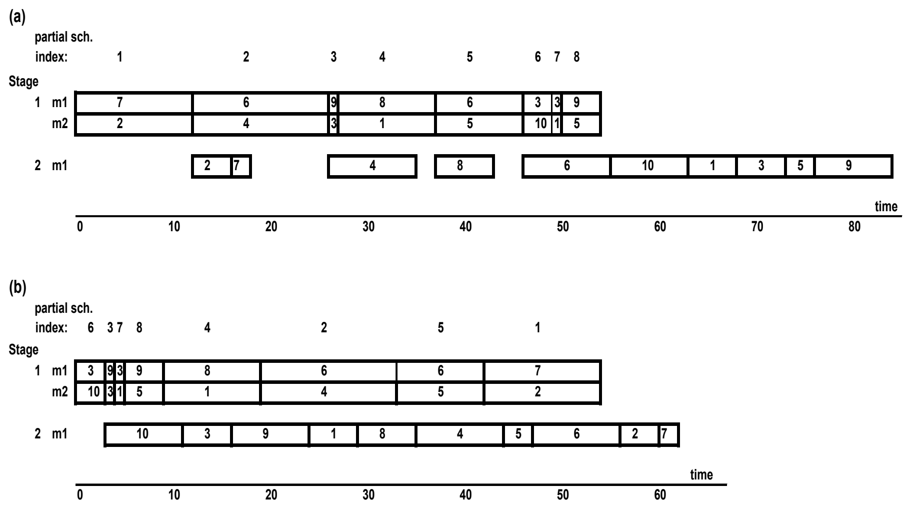

Figure 2. An illustrative example. The resulting schedules: a) the feasible schedule with a random sequence of partial schedules, b) the schedule with the partial schedule sequence which gives smaller makespan

2.2. Solving problem P1 – the approximate two-phase algorithm

The minimum makespan problem of resource constrained preemptive scheduling of parallel unrelated machines is solved by the approximate two-phase algorithm, proceeding as follows:

Let t_{ij} be the time of processing job j on machine i (at stage 1), and T be the time needed to finish processing all jobs at stage 1. The two-phase algorithm consists in solving first a linear programming (LP) problem for obtaining values of t_{ij}, ensuring the minimal value of T, T^*; the second phase consists in construction of a set of partial schedules based on times t_{ij}. In each partial schedule, resource constraints are fulfilled at every moment, a machine works on at most one job at a time and a job is processed on at most one machine at a time. The constructed set of partial schedules forms a schedule with makespan close to T^*, which is independent of the ordering of the partial schedules.

The first phase. In this phase, the following LP problem is solved:

$$\min T \tag{1}$$

$$\text{s.t.} \sum_{i=1}^{m} \frac{t_{ij}}{p_{ij}} = 1, \quad j = 1 \dots n, \tag{2}$$

$$\sum_{j=1}^{n} t_{ij} \leq T, \quad i = 1 \dots m, \tag{3}$$

$$\sum_{i=1}^{m} t_{ij} \leq T, \quad j = 1 \dots n, \tag{4}$$

$$\sum_{j \in N_r} \sum_{i=1}^{m} t_{ij} \leq W_r T, \quad r = 1 \dots l, \tag{5}$$

$$t_{ij} \geq 0, \quad i = 1 \dots m, \quad j = 1 \dots n, \tag{6}$$

$$T \geq 0, \tag{7}$$

where t_{ij}, T are decision variables, and N_r is the set of jobs that use resource of type r.

Constraints (2) ensure that the processing of each job at stage 1 is finished. Due to constraints (3) and (4), respectively, the total working time of each machine does not exceed T and each job is completed by time T. Constraints (5) say that the total usage of every resource r over time period T does not exceed the total availability of this resource ($W_r T$) over T.

As mentioned earlier, the first phase is the same as that in Słowiński (1981).

The second phase. Having the optimal values of t_{ij}, t^*_{ij}, and the optimal value of T, T^*, a schedule composed of partial schedules is constructed. The schedule construction process can be described as follows.

Step 1. Set the partial schedule index $b = 1$, $T = T^*$ and $t_{ij} = t^*_{ij}$ for $(i = 1, \ldots, m,\ j = 1, \ldots, n)$.

Step 2. Calculate criticality ratios of machines, $\varepsilon^I_i = \sum_{j=1}^n t_{ij}/T$, jobs, $\varepsilon^{II}_j = \sum_{i=1}^m t_{ij}/T$, and resources, $\varepsilon^{III}_r = \sum_{j\in N_r}\sum_{i=1}^m t_{ij}/W_r T$. Machine i, job j, and resource r are called critical if, respectively, $\varepsilon^I_i = 1$, $\varepsilon^{II}_j = 1$, and $\varepsilon^{III}_r = 1$.

Step 3. For each pair (i, j), calculate the sum of the criticality ratios. Choose a set P of at most m pairs (i, j) (for which $t_{ij} > 0$) with the greatest values of these sums so that no two pairs contain the same job index or the same machine index and resource constraints at every moment are satisfied. Pairs (i, j), chosen to be included in P, determine the assignment of machines to jobs in partial schedule S^b.

Step 4. Calculate the partial schedule length Δ_b:

Step 4.1. Let $x = \min_{(i,j)\in P}\{t_{ij}\}$,
minimum slack time of machines that do not work in S^b
$y^I = \min_{i:\varepsilon^I_i<1}\left\{T(1-\varepsilon^I_i)\right\}$,
minimum slack time of jobs that are not processed in S^b
$y^{II} = \min_{j:\varepsilon^{II}_j<1}\left\{T(1-\varepsilon^{II}_j)\right\}$,
and minimum slack time for resources that are used in S^b in quantities smaller than W_r,
$y^{III} = \min_{r:\varepsilon^{III}_r<1}\left\{W_r T(1-\varepsilon^{III}_r)/\left(W_r - \sum_{(i,j)\in P: j\in N_r} 1\right)\right\}$.

Step 4.2. If $b \le n$ set Δ_b=$\min\{x, y^I, y^{II}, y^{III}\}$, if $b > n$ set Δ_b=x.

Step 5. Decrease t_{ij} by Δ_b for $(i, j) \in P$. If $t_{ij} = 0$ for $(i = 1, \ldots, m, j = 1, \ldots, n)$ then stop, otherwise calculate

$$T = \max\left\{\max_{i=1,\ldots,m}\left\{\sum_{j=1}^n t_{ij}\right\}, \max_{j=1,\ldots,n}\left\{\sum_{i=1}^m t_{ij}\right\}, \max_{r=1,\ldots,l}\left\{\sum_{i=1}^m\sum_{j\in N_r} t_{ij}/W_r\right\}\right\},$$

set $b = b + 1$ and go to Step 2.

The second phase differs from that in the optimal two-phase method (Słowiński, 1981) as follows. In the optimal two-phase method, in Step 3, set P is constructed so that each critical machine (with $\varepsilon^I_i = 1$) works, each critical job (with $\varepsilon^{II}_j = 1$) is processed, each critical resource (with $\varepsilon^{III}_r = 1$) is used. This guarantees that the schedule has length of T^* time units. In Step 4.2, $\Delta_b = \min\{x, y^I, y^{II}, y^{III}\}$. In Step 5, T and t_{ij} (such that $(i, j) \in P$) are decreased by Δ_b, and, if $T = 0$ the algorithm stops, otherwise it goes to Step 2.

The here used approximate two-phase algorithm is faster than the optimal one. It provides solutions with the average (over all problems examined in Section 4) deviation from the optimal makespan equal to 0.08%, which does not influence the final results for the flowshop.

2.3. Solving problem P2 – the sequencing procedures

After finding a set of partial schedules, the aim is to find a sequence of these partial schedules, yielding the flowshop schedule with makespan as small as possible.

For this purpose, we propose sequencing procedures, in which first, the values of factors, denoted by e_1^b and e_2^b, are determined for each partial schedule S^b and then, the partial schedules are ordered taking into account these values. The values of e_1^b and e_2^b for a partial schedule depend on the processing times at stage 1 and/or at stage 2 of jobs which belong to this partial schedule. The processing time of job j at stage 1 (denoted by u_j) is equal to the sum of the durations of partial schedules, in which this job (or its parts) is executed ($u_j = \sum_{b:j\in F^b} \Delta_b$, where F^b is a set of indices of jobs that are processed, or whose parts are processed, in partial schedule S^b). Let us remind that the processing time of job j at stage 2 is denoted s_j. While determining the values of e_1^b and e_2^b for partial schedule S^b, we use the minimum processing time ($\min_{j\in F^b}\{u_j\}$), the maximum processing time ($\max_{j\in F^b}\{u_j\}$), the sum of processing times ($\sum_{j\in F^b} u_j$), and the average processing time ($\sum_{j\in F^b} u_j)/k^b$, where k^b is the number of jobs processed in S^b) at stage 1 of jobs from S^b, and the minimum processing time ($\min_{j\in F^b}\{s_j\}$), the maximum processing time ($\max_{j\in F^b}\{s_j\}$) and the sum of processing times ($\sum_{j\in F^b} s_j$) at stage 2 of jobs from S^b.

In the proposed sequencing procedures, popular priority rules (SPT, LPT and the Johnson algorithm, Johnson, 1954) are adapted for the situation where jobs are bound to partial schedules. In this situation sequencing the jobs is not performed directly but by means of sequencing the partial schedules.

We design twelve different sequencing procedures. For each of them, the expressions according to which the values of e_1^b and e_2^b are calculated and the way of finding the ordering of the partial schedules using these values are shown in Table 1.

The twelve sequencing procedures listed in Table 1 are used in heuristic algorithms A1-A12, examined in Section 4.

3. Lower bounds

Since determination of the optimal solution to the considered problem is practically impossible, we derive two lower bounds on the value of the optimal makespan, which will be used in evaluating the performance of the proposed heuristic algorithms.

An immediate lower bound is given by:

$$LB_1 = \sum_{j=1}^{n} s_j + \min_{\substack{i=1,\dots,m \\ j=1,\dots,n}} \{p_{ij}\}. \tag{8}$$

The first term in LB_1 is equal to the sum of job processing times at stage 2.

Table 1. The sequencing procedures

Sqeuencing procedure	Description of the sequencing procedure
SQ1	arranges partial schedules in non-decreasing order of e_1^b, where $e_1^b = \min_{j\in F^b}\{u_j\}$
SQ2	arranges partial schedules in non-decreasing order of e_1^b, where $e_1^b = \sum_{j\in F^b} u_j$
SQ3	arranges partial schedules in non-decreasing order of e_1^b, where $e_1^b = \min_{j\in F^b}\{u_j\}/\min_{j\in F^b}\{s_j\}$
SQ4	arranges partial schedules in non-decreasing order of e_1^b, where $e_1^b = \max_{j\in F^b}\{u_j\}/\max_{j\in F^b}\{s_j\}$
SQ5	arranges partial schedules in non-decreasing order of e_1^b, where $e_1^b = \sum_{j\in F^b} u_j/\sum_{j\in F^b} s_j$
SQ6	arranges partial schedules in non-decreasing order of e_1^b, where $e_1^b = \min_{j\in F^b}\{u_j\}/\max_{j\in F^b}\{s_j\}$
SQ7	arranges partial schedules in non-decreasing order of e_1^b, where $e_1^b = \min_{j\in F^b}\{u_j\}/\sum_{j\in F^b} s_j$
SQ8	arranges partial schedules in non-decreasing order of e_1^b, where $e_1^b = \sum_{j\in F^b} u_j/\max_{j\in F^b}\{s_j\}$
SQ9	arranges partial schedules in non-decreasing order of e_1^b, where $e_1^b = \min_{j\in F^b}\{u_j/s_j\}$
SQ10	first, arranges partial schedules with $e_1^b \le e_2^b$ in non-decreasing order of e_1^b, and then arranges partial schedules with $e_1^b > e_2^b$ in non-increasing order of e_2^b, where e_1^b=$\min_{j\in F^b}\{u_j\}$, e_2^b=$\max_{j\in F^b}\{s_j\}$
SQ11	first, arranges partial schedules with $e_1^b \le e_2^b$ in non-decreasing order of e_1^b, and then arranges partial schedules with $e_1^b > e_2^b$ in non-increasing order of e_2^b, where $e_1^b = \left(\sum_{j\in F^b} u_j\right)/k^b$, $e_2^b = \max_{j\in F^b}\{s_j\}$
SQ12	first, arranges partial schedules with $e_1^b \le e_2^b$ in non-decreasing order of e_1^b, and then arranges partial schedules with $e_1^b > e_2^b$ in non-increasing order of e_2^b, where $e_1^b = \left(\sum_{j\in F^b} u_j\right)/k^b$, $e_2^b = \sum_{j\in F^b} s_j$

F^b is a set of indices of jobs that are processed (or whose parts are processed) in partial schedule S^b, u_j is the total processing time of job j at stage 1, k^b is the number of jobs processed in S^b.

The second term is equal to the smallest job processing time at stage 1 (the machine at stage 2 remains idle for at least the time needed to finish at stage 1 processing a job with the smallest processing time).

Another lower bound is given by:

$$LB_2 = C_1^* + \min_{j=1,\dots,n} \{s_j\}, \tag{9}$$

where C_1^* denotes the minimal makespan (i.e. the minimum time needed to finish processing all jobs at stage 1) for the problem occurring at stage 1 ($C_1^*{=}T^*$). The second term represents the minimum unavoidable idleness at stage 1 which is equal to the smallest job processing time at stage 2.

LB_1 and LB_2 will be effective for problem instances which are dominated by jobs with large processing times at stage 2 and stage 1, respectively.

Hence, a lower bound on the optimal makespan in the considered two-stage flowshop will be:

$$LB = \max\{LB_1, LB_2\}. \tag{10}$$

4. Computational experiment

An extensive computational experiment was carried out to evaluate the performance of the proposed heuristic algorithms as well as to determine the effect of job characteristics and the numbers of jobs, machines and resources on the effectiveness of the algorithms.

The number of jobs was considered to be n =100, 300 and 600.

The number of machines, m, at stage 1 was set at 2, 4 and 6.

We considered the following two ways of modeling resource constraints:

– the number of resource types l was set to 1, the resource availability W_1 was set at $m/2$ and resource requirements were set at 1 for 75% of jobs, the rest of jobs did not require the resources;

– the number of resource types l was set at 2, 4, and 6 for problems with, respectively, 2, 4 and 6 machines, the resource availability W_r was set at 1 for each resource type ($r = 1,\dots,l$), each job required 1 unit of one randomly chosen resource type.

The job processing times at stage 2 were generated from $U[1,\ 100]$ ($U[a,\ b]$ denotes the discrete uniform distribution in the range of $[a,\ b]$), the processing times at stage 1 were generated from 10 intervals for each problem size.

For each examined combination of n, m, l and the processing time interval, 50 problems were generated. This results in 9000 problem instances. These randomly generated instances were used to test the average performance of the heuristic algorithms.

To evaluate the effectiveness of the proposed algorithms we used the value of the percentage deviation of the heuristic makespan from the lower bound on the optimal makespan:

$$\delta = \frac{C - LB}{LB} \times 100\%, \tag{11}$$

C is the makespan found by a heuristic algorithm.

For each problem size and each processing time interval the ratio of the minimal time needed for finishing processing all jobs at stage 1 to the sum of job processing times at stage 2, denoted by θ ($\theta = C_1^* / \sum_1^n s_j$), was calculated. Effectiveness of the proposed heuristic algorithms was analyzed in terms of θ. Due to the generation of job processing times at stage 1 from 10 intervals we can consider the cases with θ less than 1 (stage 2 is dominant), with θ close to 1 (the stages are balanced) and with θ greater than 1 (stage 1 is dominant).

Another performance measure is the CPU time (reported in seconds) consumed by the heuristic algorithms.

All programs for the algorithms presented in the paper were written in C++ and run on a PC computer with Celeron 1.3 GHz. The LP problem (1-7) was solved using lp_solve optimizer v.5.5 available from http://groups.yahoo.com/-group/lp_solve.

The results of the computational experiment are presented in Tables 2-8. In Tables 2-7, the average percentage deviations of the heuristic makespan from the lower bound on the optimal makespan are shown for the proposed algorithms A1-A12 and for algorithm (ARAND), that creates a random sequence of partial schedules at stage 1. In these tables, the average values of θ calculated for each problem size and each processing time interval are also depicted (column 3). Table 8 contains computation times.

In Tables 2-7 we can see that all the algorithms A1-A12 produce significantly better results than algorithm ARAND and that different algorithms are the best for different problems (the best results are given in bold).

For problems with two machines (Tables 2 and 5) several algorithms (A4-A12) provide excellent quality results (with δ close to 0), but algorithms A10 and A11 are superior to the others. The average δ (over all the instances with m=2, Tables 2 and 5) equals 0.01%, 0.01%, 0.02%, 0.02%, 0.02%, 0.02%, 0.03%, 0.03%, 0.03%, 0.13%, 0.34% and 0.42%, respectively, for algorithms A10, A11, A6, A7, A9, A12, A4, A5, A8, A3, A1 and A2. The average δ of ARAND is 2.01%.

For problems with bigger number of machines (m = 4 and 6, Tables 3, 4, 6 and 7) algorithm A9 is the best. The next best algorithm is A6. For problems with 4 machines the average δ (over all the instances with m=4, Tables 3 and 6) equals 0.39%, 0.47%, 0.53%, 0.54%, 0.59%, 0.71%, 0.89%, 1.03%, 1.29%, 1.36%, 1.39% and 1.88% for algorithms A9, A6, A5, A7, A10, A8, A1, A3, A4, A11, A2 and A12, respectively. The average δ of ARAND is 9.89%. For problems with 6 machines the average δ (over all the instances with m=6, Tables 4 and 7) is 1.11%, 1.33%, 1.60%, 1.61%, 2.05%, 2.47%, 2.81%, 3.09%, 3.36%, 5.47%, 5.87% and 7.65% for algorithms A9, A6, A1, A7, A5, A8, A10, A2, A3, A4, A11 and A12. The average δ of ARAND is 17.81%.

An interesting behavior of algorithms can be observed for problems with 4 and 6 machines and more than 1 resource type (Tables 6 and 7). In this case, algorithm A9 gives the best results in the region where θ is less than or close to

1, while in the region with greater values of θ algorithm A5 becomes superior to other algorithms.

Most of the algorithms exhibit slightly worse performance for problems with a greater number of resource types. The average deviations δ obtained by algorithms A1-A12 are respectively equal to 0.85%, 1.64%, 1.04%, 1.64%, 0.80%, 0.46%, 0.45%, 0.99%, 0.38%, 0.67%, 1.57% and 2.07% for problems with 1 resource type (Tables 2, 3 and 4), and 1.03%, 1.62%, 1.98%, 2.89%, 0.94%, 0.75%, 1.00%, 1.15%, 0.63%, 1.61%, 3.26% and 4.29% for problems with more than 1 resource type (Tables 5, 6 and 7).

The most difficult problems (with the greatest value of δ) are the ones with θ close to 1. In the region of θ close to 1 the stages in the flowshop are balanced (i.e. the minimum makespan of the schedule at stage 1 is close to the sum of the job processing times at stage 2) and the total idle time (the sum of the idle time at stage 1, i.e. time when no machine works, and the idle time of the machine at stage 2) is small, hence this region is very important from the practical point of view. However, when θ is close to 1, for all problem sizes, one can see high peaks in the values of deviation δ obtained by algorithm ARAND. The proposed algorithms significantly lower these peaks. On the average (over all instances considered), the ratio of the maximal deviation from algorithm ARAND for problems with given n, m and l to the average maximal deviation provided by the best algorithm for the same problems equals 402.53, 31.04 and 15.29, respectively, for problems with 2, 4, and 6 machines.

The performance of all algorithms A1-A12 improves considerably when the number of jobs grows. When the number of machines increases the performance of the algorithms deteriorates. This is caused by the fact that the greater the number of machines the more jobs are tied to a partial schedule and it is more difficult to find an appropriate ordering of jobs ensuring a short schedule in the whole system. The increase in the values of δ caused by the increase in the number of resources can be explained by the fact that when the number of resources grows the resource constraints become weaker and more jobs are executed at the same time, so that finding a proper sequence of jobs is more difficult.

Table 2. Computational results for problems with the number of machines m=2 and 1 resource type

n	p_{ij}	θ	$\delta\%$												
			A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	ARAND
100	[1,100]	0.50	**0.00**	0.01	0.01	0.01	0.01	**0.00**	0.01	0.01	0.01	**0.00**	0.01	0.01	1.12
	[1,200]	1.03	1.20	3.27	0.25	0.01	0.01	0.01	0.01	0.02	0.01	**0.00**	0.01	**0.00**	7.54
	[1,300]	1.50	1.10	1.39	0.35	0.03	0.03	0.02	0.02	0.05	0.03	0.01	0.01	0.01	2.54
	[1,400]	1.94	0.87	1.15	0.21	0.04	0.04	0.02	0.02	0.05	0.04	0.01	0.01	0.01	1.22
	[1,500]	2.46	0.56	0.85	0.22	0.01	0.01	0.01	0.01	0.02	0.01	**0.00**	**0.00**	**0.00**	0.88
	[1,600]	2.97	0.63	0.70	0.16	0.02	0.02	0.01	0.01	0.03	0.02	**0.00**	**0.00**	**0.00**	0.77
	[1,700]	3.46	0.45	0.54	0.12	0.05	0.04	0.02	0.01	0.06	0.02	0.01	0.01	0.01	0.57
	[1,800]	3.94	0.33	0.51	0.13	0.02	0.03	0.01	0.01	0.03	0.02	0.01	0.01	0.01	0.39
	[1,900]	4.47	0.27	0.44	0.09	0.01	0.01	0.01	0.01	0.01	0.01	**0.00**	**0.00**	**0.00**	0.39
	[1,1000]	4.95	0.26	0.39	0.08	0.02	0.01	**0.00**	**0.00**	0.02	0.01	**0.00**	**0.00**	**0.00**	0.31
		average	0.57	0.92	0.16	0.02	0.02	0.01	0.01	0.03	0.02	**0.00**	0.01	0.01	1.57
300	[1,100]	0.51	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.41
	[1,200]	0.99	0.31	0.85	0.09	0.01	0.01	**0.00**	**0.00**	0.02	**0.00**	**0.00**	**0.00**	**0.00**	6.51
	[1,300]	1.49	0.50	0.44	0.09	0.02	0.01	**0.00**	**0.00**	0.02	0.01	**0.00**	**0.00**	**0.00**	0.86
	[1,400]	2.00	0.33	0.36	0.07	0.04	0.02	**0.00**	**0.00**	0.03	0.01	**0.00**	**0.00**	**0.00**	0.48
	[1,500]	2.50	0.21	0.24	0.05	0.01	0.01	**0.00**	**0.00**	0.01	**0.00**	**0.00**	**0.00**	**0.00**	0.23
	[1,600]	3.00	0.17	0.20	0.03	0.01	0.01	**0.00**	**0.00**	0.01	**0.00**	**0.00**	**0.00**	**0.00**	0.17
	[1,700]	3.46	0.14	0.19	0.04	0.01	0.01	**0.00**	**0.00**	0.01	0.01	**0.00**	**0.00**	**0.00**	0.16
	[1,800]	4.01	0.12	0.17	0.04	0.01	0.01	**0.00**	**0.00**	0.01	0.01	**0.00**	**0.00**	**0.00**	0.15
	[1,900]	4.46	0.11	0.14	0.03	0.01	0.01	**0.00**	**0.00**	0.01	0.01	**0.00**	**0.00**	**0.00**	0.12
	[1,1000]	4.96	0.09	0.12	0.03	0.01	**0.00**	**0.00**	**0.00**	0.01	**0.00**	**0.00**	**0.00**	**0.00**	0.10
		average	0.20	0.27	0.05	0.01	0.01	**0.00**	**0.00**	0.01	0.01	**0.00**	**0.00**	**0.00**	0.92
600	[1,100]	0.50	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.25
	[1,200]	0.99	0.19	0.73	0.04	0.01	0.01	**0.00**	**0.00**	0.01	**0.00**	**0.00**	0.04	**0.00**	6.47
	[1,300]	1.51	0.25	0.22	0.06	0.02	0.01	**0.00**	0.01	0.02	**0.00**	**0.00**	**0.00**	**0.00**	0.50
	[1,400]	1.99	0.16	0.17	0.04	0.01	0.01	**0.00**	**0.00**	0.01	0.01	**0.00**	**0.00**	**0.00**	0.19
	[1,500]	2.46	0.13	0.13	0.03	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.12
	[1,600]	3.06	0.09	0.10	0.02	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.08
	[1,700]	3.48	0.08	0.10	0.03	0.01	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.06
	[1,800]	3.99	0.07	0.08	0.02	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.07
	[1,900]	4.49	0.06	0.07	0.02	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.05
	[1,1000]	4.99	0.04	0.05	0.01	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.05
		average	0.11	0.17	0.03	0.01	**0.00**	**0.00**	**0.00**	0.01	**0.00**	**0.00**	**0.00**	**0.00**	0.78

Table 3. Computational results for problems with the number of machines m=4 and 1 resource type

n	p_{ij}	θ	δ%												
			A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	ARAND
100	[1,400]	0.60	**0.00**	0.67	0.03	0.75	0.37	**0.00**	0.01	0.54	0.01	**0.00**	0.54	0.38	8.83
	[1,600]	0.88	0.85	3.61	0.79	2.54	1.08	0.16	0.16	1.49	0.14	**0.12**	1.89	2.20	21.44
	[1,800]	1.19	3.30	6.05	4.09	3.98	2.32	1.54	1.45	3.48	**1.18**	1.54	3.25	3.96	20.54
	[1,1000]	1.48	2.63	3.23	2.23	2.24	**0.92**	1.39	1.36	1.54	1.08	1.28	2.06	2.87	13.41
	[1,1200]	1.72	2.12	2.53	1.63	1.59	**0.83**	1.09	1.07	1.24	0.89	1.06	1.40	1.91	9.05
	[1,1400]	2.03	1.81	2.24	1.29	1.00	**0.73**	0.93	0.96	1.06	0.74	0.74	0.91	1.27	6.27
	[1,1600]	2.38	1.47	1.95	1.22	0.63	**0.50**	0.70	0.62	0.85	0.63	0.56	0.78	0.92	3.90
	[1,1800]	2.79	1.30	1.69	1.17	0.85	**0.50**	0.67	0.65	0.79	0.53	0.57	0.60	0.75	3.34
	[1,2000]	3.02	1.24	1.49	0.96	0.70	0.53	0.62	0.55	0.72	0.56	**0.51**	0.56	0.54	2.67
	[1,2200]	3.22	1.13	1.34	0.81	0.51	**0.39**	0.53	0.53	0.58	0.42	0.42	0.48	0.44	2.23
		average	1.59	2.48	1.42	1.48	0.82	0.76	0.74	1.23	**0.62**	0.68	1.25	1.52	9.17
300	[1,400]	0.60	**0.00**	0.32	**0.00**	0.32	0.27	**0.00**	**0.00**	0.29	**0.00**	**0.00**	0.05	0.17	6.48
	[1,600]	0.90	0.26	2.03	0.26	1.65	1.18	0.10	0.15	1.27	**0.08**	0.10	0.59	0.97	19.84
	[1,800]	1.18	1.21	3.00	1.27	1.65	0.91	0.34	0.45	1.06	**0.26**	0.35	1.23	1.91	18.04
	[1,1000]	1.47	0.97	1.43	0.72	0.83	0.32	0.30	0.32	0.38	**0.24**	**0.24**	0.40	0.60	9.22
	[1,1200]	1.77	0.68	1.09	0.50	0.38	0.24	0.17	0.19	0.36	0.16	**0.15**	0.21	0.22	4.81
	[1,1400]	2.10	0.54	0.74	0.39	0.31	0.18	0.15	0.17	0.20	0.13	**0.12**	0.13	0.17	2.76
	[1,1600]	2.37	0.58	0.73	0.39	0.38	**0.17**	0.20	0.25	0.25	**0.17**	0.25	0.29	0.31	1.83
	[1,1800]	2.67	0.46	0.55	0.30	0.26	0.15	0.12	0.13	0.22	0.14	**0.08**	0.09	0.09	1.28
	[1,2000]	2.97	0.43	0.60	0.34	0.33	0.14	0.13	0.13	0.20	0.15	**0.12**	**0.12**	0.15	0.96
	[1,2200]	3.23	0.38	0.44	0.28	0.21	0.11	0.12	0.15	0.16	0.11	**0.10**	0.11	0.14	1.08
		average	0.55	1.09	0.45	0.63	0.37	0.16	0.19	0.44	**0.14**	0.15	0.32	0.47	6.63
600	[1,400]	0.60	**0.00**	0.35	**0.00**	0.45	0.32	**0.00**	**0.00**	0.32	**0.00**	**0.00**	0.04	0.06	5.62
	[1,600]	0.88	0.00	1.19	0.05	0.84	0.75	**0.00**	**0.00**	0.81	**0.00**	**0.00**	0.81	0.57	17.73
	[1,800]	1.18	0.69	1.80	0.62	0.65	0.64	0.16	0.26	0.51	**0.15**	0.34	0.88	1.33	15.96
	[1,1000]	1.48	0.46	0.85	0.33	0.40	0.11	0.10	0.12	0.17	0.10	**0.07**	0.11	0.29	8.13
	[1,1200]	1.78	0.35	0.55	0.22	0.12	0.06	**0.05**	0.06	0.09	0.06	**0.05**	0.05	0.05	3.88
	[1,1400]	2.08	0.35	0.53	0.19	0.20	0.13	0.05	0.05	0.16	0.05	**0.03**	0.03	0.06	1.80
	[1,1600]	2.37	0.29	0.33	0.16	0.13	0.05	0.06	0.07	0.08	**0.04**	0.10	0.12	0.13	1.23
	[1,1800]	2.67	0.24	0.30	0.16	0.45	**0.09**	0.12	0.15	0.13	0.10	0.19	0.22	0.19	0.97
	[1,2000]	2.97	0.22	0.27	0.16	0.12	0.06	0.06	0.06	0.08	0.06	**0.05**	**0.05**	**0.05**	0.56
	[1,2200]	3.23	0.19	0.31	0.13	0.17	0.06	0.05	0.06	0.11	0.05	**0.03**	**0.03**	**0.03**	0.54
		average	0.28	0.65	0.20	0.35	0.23	0.07	0.08	0.25	**0.06**	0.09	0.23	0.27	5.64

Table 4. Computational results for problems with the number of machines m=6 and 1 resource type

n	p_{ij}	θ	$\delta\%$												
			A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	ARAND
100	[1,600]	0.44	**0.00**	0.82	0.05	1.39	0.49	**0.00**	0.01	0.78	0.01	**0.00**	0.92	0.53	6.23
	[1,900]	0.65	**0.00**	2.16	0.74	4.15	1.51	0.02	0.02	1.76	0.01	**0.00**	3.43	2.50	17.13
	[1,1200]	0.88	1.47	6.48	5.03	10.76	3.55	1.18	0.88	5.03	**0.59**	1.78	10.62	11.64	28.06
	[1,1500]	1.03	4.76	10.22	9.18	14.48	7.23	3.73	3.49	9.55	**3.09**	7.05	16.17	16.68	33.23
	[1,1800]	1.24	4.63	7.67	7.17	10.45	4.77	3.81	**3.51**	6.29	3.90	6.50	11.17	14.32	27.86
	[1,2100]	1.51	4.00	5.33	5.61	6.80	3.29	3.30	3.36	4.44	**2.94**	5.11	7.05	9.07	18.71
	[1,2400]	1.72	3.43	4.89	4.20	4.03	2.29	2.74	2.54	3.23	**2.15**	3.18	4.45	5.86	14.78
	[1,2700]	1.91	2.59	3.55	3.03	4.22	1.97	2.09	2.07	2.80	**1.93**	2.68	3.53	4.69	11.80
	[1,3000]	2.08	2.59	3.38	2.96	2.52	**1.43**	1.92	1.65	2.16	1.47	1.74	2.23	3.50	9.59
	[1,3300]	2.35	2.37	2.59	2.22	2.28	**1.30**	1.68	1.44	1.84	1.37	1.83	2.45	2.92	8.42
		average	2.58	4.71	4.02	6.11	2.78	2.05	1.90	3.79	**1.75**	2.99	6.20	7.17	17.58
300	[1,600]	0.43	**0.00**	0.47	0.01	0.47	0.29	**0.00**	**0.00**	0.34	**0.00**	**0.00**	0.19	0.28	4.50
	[1,900]	0.64	**0.00**	0.95	0.01	1.33	0.99	**0.00**	**0.00**	0.85	**0.00**	**0.00**	1.15	0.80	14.44
	[1,1200]	0.85	0.17	4.96	1.12	7.41	3.40	0.10	**0.06**	3.92	0.10	0.31	6.57	6.92	25.53
	[1,1500]	1.09	2.89	7.78	6.62	9.48	5.17	1.67	2.00	5.86	**1.28**	3.88	10.22	14.64	28.89
	[1,1800]	1.29	2.12	4.15	3.89	5.36	2.58	1.51	1.46	2.83	**1.29**	2.82	6.21	10.48	21.27
	[1,2100]	1.49	1.60	3.00	2.11	3.75	1.99	1.09	1.20	1.98	**0.91**	1.82	3.33	5.31	16.11
	[1,2400]	1.71	1.41	1.92	1.54	2.50	1.15	0.97	0.91	1.24	**0.72**	1.17	2.04	3.72	10.97
	[1,2700]	1.91	1.18	1.68	1.07	2.11	0.99	0.73	**0.70**	1.06	0.60	0.97	1.54	2.11	8.86
	[1,3000]	2.15	1.01	1.25	0.88	1.42	**0.59**	0.72	0.60	0.85	0.69	0.77	1.23	1.69	6.52
	[1,3300]	2.32	0.85	1.03	0.74	1.03	0.52	**0.43**	0.50	0.58	0.45	0.52	0.71	1.30	5.00
		average	1.12	2.72	1.80	3.49	1.77	0.72	0.74	1.95	**0.60**	1.23	3.32	4.72	14.21
600	[1,600]	0.43	**0.00**	0.40	**0.00**	0.45	0.29	**0.00**	**0.00**	0.33	**0.00**	**0.00**	0.15	0.19	4.48
	[1,900]	0.64	**0.00**	0.66	**0.00**	1.20	0.47	**0.00**	**0.00**	0.57	**0.00**	**0.00**	0.71	0.42	12.92
	[1,1200]	0.86	0.12	2.27	0.74	4.93	1.40	0.02	0.01	1.56	**0.00**	0.18	6.26	6.28	24.68
	[1,1500]	1.06	1.72	6.09	5.77	8.72	3.72	0.88	1.04	4.06	**0.63**	3.74	10.70	15.57	30.69
	[1,1800]	1.29	1.30	3.09	2.27	4.67	2.28	0.67	0.67	2.06	**0.43**	1.43	4.47	8.41	20.09
	[1,2100]	1.49	1.01	1.68	0.99	2.07	1.02	0.47	0.45	1.09	**0.31**	1.02	2.23	5.57	14.23
	[1,2400]	1.69	0.86	1.36	0.83	1.89	0.99	0.45	0.59	0.81	**0.32**	0.68	1.55	3.21	10.31
	[1,2700]	1.90	0.67	1.02	0.59	1.43	0.85	0.40	0.44	0.64	**0.36**	0.74	1.22	2.70	7.42
	[1,3000]	2.12	0.54	0.66	0.42	0.48	0.50	0.31	0.43	0.41	**0.29**	0.38	0.58	1.41	5.54
	[1,3300]	2.33	0.61	0.72	0.41	0.72	0.31	0.30	0.39	0.34	**0.27**	0.37	0.45	0.89	4.70
		average	0.68	1.79	1.20	2.65	1.18	0.35	0.40	1.19	**0.26**	0.85	2.83	4.46	13.51

Table 5. Computational results for problems with the number of machines m=2 and the number of resource types $l = 2$

n	p_{ij}	θ	$\delta\%$												
			A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	ARAND
100	[1,100]	0.36	**0.00**	0.03	0.01	0.04	0.04	**0.00**	0.01	0.04	0.01	**0.00**	0.03	0.03	1.04
	[1,200]	0.73	**0.00**	0.10	**0.00**	0.07	0.06	**0.00**	**0.00**	0.07	0.00	**0.00**	0.05	0.04	7.37
	[1,300]	1.07	1.58	1.44	0.84	0.18	0.17	0.12	0.24	0.21	0.13	**0.11**	0.13	0.48	14.48
	[1,400]	1.49	1.41	1.50	0.63	0.14	0.14	0.14	0.14	0.16	0.15	**0.05**	**0.05**	0.08	5.95
	[1,500]	1.84	1.03	1.05	0.45	0.11	0.10	0.09	0.08	0.18	0.09	**0.07**	0.08	0.10	3.17
	[1,600]	2.23	0.85	0.86	0.45	0.13	0.11	0.11	0.18	0.11	0.11	**0.06**	**0.06**	0.08	1.88
	[1,700]	2.57	0.82	0.82	0.36	0.14	0.09	0.08	0.13	0.09	0.09	**0.06**	**0.06**	0.09	1.35
	[1,800]	2.84	0.68	0.75	0.32	0.07	0.07	0.06	0.05	0.11	0.05	**0.04**	**0.04**	**0.04**	1.29
	[1,900]	3.29	0.50	0.57	0.23	0.07	0.06	0.04	0.05	0.06	0.05	**0.03**	**0.03**	**0.03**	1.10
	[1,1000]	3.64	0.52	0.60	0.26	0.06	0.05	0.06	0.07	0.08	0.05	**0.03**	**0.03**	**0.03**	1.02
		average	0.74	0.77	0.36	0.10	0.09	0.07	0.10	0.11	0.07	**0.04**	0.05	0.10	3.86
300	[1,100]	0.35	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.30
	[1,200]	0.70	**0.00**	0.01	**0.00**	0.01	0.01	**0.00**	**0.00**	0.01	**0.00**	**0.00**	0.01	0.01	4.24
	[1,300]	1.06	0.64	0.57	0.28	0.08	0.05	0.03	0.05	0.08	0.04	**0.02**	0.03	0.04	13.09
	[1,400]	1.40	0.49	0.47	0.22	0.06	0.05	0.04	0.05	0.04	0.05	**0.02**	**0.02**	0.04	4.46
	[1,500]	1.72	0.46	0.41	0.18	0.03	0.02	0.02	0.03	0.03	0.02	**0.01**	**0.01**	0.03	1.82
	[1,600]	2.09	0.34	0.31	0.17	0.03	0.02	0.03	0.03	0.03	0.03	**0.01**	**0.01**	0.02	0.89
	[1,700]	2.44	0.30	0.27	0.12	0.03	0.02	0.02	0.02	0.03	0.02	**0.01**	**0.01**	**0.01**	0.75
	[1,800]	2.81	0.22	0.20	0.11	0.02	0.02	0.02	0.02	0.02	0.02	**0.01**	**0.01**	**0.01**	0.53
	[1,900]	3.13	0.21	0.22	0.09	0.02	0.02	**0.01**	0.02	0.02	0.02	**0.01**	**0.01**	**0.01**	0.37
	[1,1000]	3.51	0.17	0.19	0.09	0.01	0.01	0.01	0.01	0.01	0.01	**0.00**	**0.00**	0.01	0.40
		average	0.28	0.27	0.13	0.03	0.02	0.02	0.02	0.03	0.02	**0.01**	**0.01**	0.02	2.69
600	[1,100]	0.35	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.16
	[1,200]	0.69	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	3.04
	[1,300]	1.03	0.28	0.26	0.12	0.02	**0.01**	**0.01**	**0.01**	0.02	**0.01**	**0.01**	**0.01**	0.02	12.29
	[1,400]	1.38	0.28	0.25	0.11	0.02	**0.01**	**0.01**	0.02	0.02	**0.01**	**0.01**	**0.01**	**0.01**	3.83
	[1,500]	1.72	0.21	0.18	0.08	0.02	0.01	0.01	0.01	0.01	0.01	**0.00**	**0.00**	0.01	1.27
	[1,600]	2.05	0.18	0.16	0.08	0.01	0.01	0.01	0.01	0.01	0.01	**0.00**	**0.00**	0.01	0.74
	[1,700]	2.44	0.16	0.15	0.07	0.01	0.01	**0.00**	0.01	0.01	0.01	**0.00**	**0.00**	**0.00**	0.35
	[1,800]	2.74	0.14	0.12	0.06	0.01	0.01	**0.00**	0.01	0.01	0.01	**0.00**	**0.00**	**0.00**	0.29
	[1,900]	3.11	0.10	0.10	0.05	0.01	0.01	**0.00**	**0.00**	0.01	0.01	**0.00**	**0.00**	**0.00**	0.24
	[1,1000]	3.43	0.09	0.09	0.05	0.01	0.01	**0.00**	**0.00**	0.01	**0.00**	**0.00**	**0.00**	**0.00**	0.21
		average	0.14	0.13	0.06	0.01	0.01	0.01	0.01	0.01	0.01	**0.00**	**0.00**	0.01	2.24

Table 6. Computational results for problems with the number of machines m=4 and the number of resource types $l = 4$

n	p_{ij}	θ	$\delta\%$												
			A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	ARAND
100	[1,400]	0.51	**0.00**	0.53	0.04	0.80	0.36	**0.00**	0.01	0.52	0.01	**0.00**	0.67	0.31	8.35
	[1,600]	0.74	0.22	2.80	0.68	3.65	0.84	0.22	0.11	1.82	**0.04**	0.23	3.10	1.55	19.77
	[1,800]	1.02	2.60	4.98	6.90	6.50	2.52	1.53	2.05	3.46	**1.18**	2.36	7.38	8.35	27.37
	[1,1000]	1.31	3.45	4.69	4.08	4.46	1.80	2.22	2.48	2.58	**1.75**	3.36	5.81	6.58	22.09
	[1,1200]	1.50	2.86	3.28	3.60	3.26	1.73	1.86	2.17	2.17	**1.71**	2.35	3.70	5.63	17.04
	[1,1400]	1.79	2.48	2.70	2.39	2.14	**1.24**	1.65	1.92	1.55	1.44	1.98	2.51	4.02	11.15
	[1,1600]	2.09	2.07	2.38	1.65	1.24	**0.89**	1.17	1.28	1.33	0.94	1.04	1.33	1.80	8.42
	[1,1800]	2.26	2.16	2.34	1.93	1.39	**1.15**	1.39	1.43	1.35	1.20	1.33	1.48	1.93	7.67
	[1,2000]	2.52	1.66	1.90	1.53	1.20	**0.83**	0.98	1.01	0.99	0.91	0.91	1.05	1.39	5.48
	[1,2200]	2.83	1.31	1.60	1.18	0.79	0.63	0.82	0.77	0.77	0.77	**0.56**	0.60	0.92	4.83
		average	1.88	2.72	2.40	2.54	1.20	1.18	1.32	1.65	**1.00**	1.41	2.76	3.25	13.22
300	[1,400]	0.46	**0.00**	0.09	0.01	0.18	0.08	**0.00**	**0.00**	0.10	**0.00**	**0.00**	0.14	0.08	5.09
	[1,600]	0.69	**0.00**	0.23	0.03	1.12	0.18	**0.00**	**0.00**	0.26	**0.00**	**0.00**	0.85	0.25	15.35
	[1,800]	0.89	0.28	1.57	1.62	4.62	0.35	0.09	0.22	0.68	**0.08**	0.28	4.83	5.38	26.01
	[1,1000]	1.15	1.37	1.86	3.53	3.88	0.96	0.92	1.47	0.99	**0.77**	2.33	5.80	9.62	24.13
	[1,1200]	1.41	1.25	1.41	1.79	2.06	**0.57**	0.77	1.01	0.75	0.68	1.36	2.48	4.65	17.00
	[1,1400]	1.63	1.01	1.05	0.98	0.90	**0.41**	0.61	0.80	0.53	0.51	0.82	1.39	2.96	11.16
	[1,1600]	1.82	0.87	0.85	0.94	0.61	**0.36**	0.52	0.72	0.43	0.44	0.55	0.83	2.05	8.53
	[1,1800]	2.03	0.83	0.73	0.67	0.65	**0.32**	0.45	0.68	0.39	0.41	0.66	0.87	1.84	6.63
	[1,2000]	2.33	0.72	0.66	0.55	0.41	**0.26**	0.38	0.48	0.31	0.36	0.33	0.37	0.91	4.00
	[1,2200]	2.55	0.68	0.66	0.52	0.43	**0.25**	0.34	0.49	0.30	0.30	0.34	0.38	1.00	3.76
		average	0.70	0.91	1.06	1.49	0.37	0.41	0.59	0.47	**0.36**	0.67	1.79	2.87	12.17
600	[1,400]	0.45	**0.00**	0.04	**0.00**	0.05	0.03	**0.00**	**0.00**	0.04	**0.00**	**0.00**	0.05	0.04	4.09
	[1,600]	0.66	**0.00**	0.14	**0.00**	0.35	0.07	**0.00**	**0.00**	0.09	**0.00**	**0.00**	0.50	0.08	13.41
	[1,800]	0.89	0.01	0.80	1.28	4.07	0.23	0.00	0.01	0.36	**0.00**	0.23	4.53	4.44	25.56
	[1,1000]	1.10	0.83	0.91	2.28	4.97	**0.40**	0.46	0.71	0.42	0.42	2.19	7.12	10.66	27.62
	[1,1200]	1.31	0.64	0.61	1.07	1.17	0.31	0.37	0.65	0.31	**0.30**	1.07	2.99	5.91	18.84
	[1,1400]	1.54	0.54	0.63	0.52	0.77	0.26	0.30	0.40	0.38	**0.25**	0.51	1.19	2.59	12.94
	[1,1600]	1.77	0.50	0.41	0.37	0.38	**0.18**	0.25	0.40	0.22	0.21	0.34	0.49	1.63	8.53
	[1,1800]	1.95	0.41	0.50	0.39	0.35	**0.17**	0.24	0.36	0.20	0.20	0.35	0.45	1.34	6.51
	[1,2000]	2.19	0.37	0.35	0.28	0.30	**0.16**	0.19	0.28	0.17	0.19	0.23	0.28	0.87	4.35
	[1,2200]	2.40	0.37	0.35	0.29	0.28	**0.17**	0.26	0.35	0.18	0.20	0.32	0.39	1.04	3.48
		average	0.37	0.47	0.65	1.27	0.20	0.21	0.32	0.24	**0.18**	0.52	1.80	2.86	12.54

Table 7. Computational results for problems with the number of machines m=6 and the number of resource types $l = 6$

n	p_{ij}	θ	$\delta\%$												
			A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	ARAND
100	[1,600]	0.39	**0.00**	0.82	0.16	1.81	0.61	0.01	0.02	0.64	0.01	**0.00**	1.15	0.63	7.36
	[1,900]	0.63	**0.01**	2.89	1.61	5.31	1.74	0.02	0.02	2.11	**0.02**	0.07	4.45	2.65	18.15
	[1,1200]	0.81	1.13	6.30	6.07	11.76	4.93	1.39	2.50	5.96	**0.76**	3.15	11.09	11.01	30.01
	[1,1500]	1.05	5.60	10.11	11.35	15.10	7.03	5.27	6.40	8.61	**4.19**	9.39	15.92	19.94	35.23
	[1,1800]	1.24	5.80	7.93	10.28	11.68	5.60	5.35	6.31	7.21	**4.37**	8.19	13.29	17.89	30.99
	[1,2100]	1.46	5.02	6.56	8.35	8.71	4.53	5.04	6.30	5.52	**4.37**	7.11	9.83	13.60	23.46
	[1,2400]	1.62	4.58	5.64	6.17	6.38	**3.43**	4.09	5.01	4.50	3.62	6.02	7.74	10.94	19.80
	[1,2700]	1.90	3.98	4.23	4.41	4.65	**2.79**	3.62	3.98	3.35	3.13	5.27	6.26	9.37	15.32
	[1,3000]	2.01	3.57	3.68	4.15	3.29	**2.21**	3.04	3.35	2.74	2.63	3.99	4.68	7.67	12.14
	[1,3300]	2.20	2.63	3.14	3.09	2.63	**1.68**	2.44	2.69	2.25	2.06	3.00	3.53	5.54	10.40
		average	3.23	5.13	5.56	7.13	3.46	3.03	3.66	4.29	**2.52**	4.62	7.79	9.92	20.29
300	[1,600]	0.36	**0.00**	0.17	0.01	0.60	0.17	**0.00**	**0.00**	0.17	**0.00**	**0.00**	0.35	0.15	4.42
	[1,900]	0.53	**0.00**	0.48	0.01	2.50	0.40	**0.00**	**0.00**	0.50	**0.00**	**0.00**	1.83	0.46	12.13
	[1,1200]	0.73	**0.00**	2.02	1.61	7.58	1.25	0.01	0.01	1.86	**0.00**	0.37	6.60	3.61	22.91
	[1,1500]	0.88	0.81	4.27	7.62	13.56	2.99	0.79	1.37	3.64	**0.45**	3.58	13.50	15.80	32.77
	[1,1800]	1.04	2.61	6.57	12.57	17.08	5.57	2.63	4.03	6.09	**2.09**	10.40	18.57	23.25	36.76
	[1,2100]	1.24	2.91	3.16	7.01	9.88	2.28	2.64	4.24	2.92	**2.24**	7.94	12.59	17.77	28.51
	[1,2400]	1.38	2.20	2.35	5.60	6.62	2.01	2.27	3.36	2.04	**1.83**	6.21	9.67	14.92	24.00
	[1,2700]	1.61	1.79	1.90	4.02	3.75	1.45	1.52	2.21	1.48	**1.23**	3.73	5.82	10.02	18.24
	[1,3000]	1.79	1.58	1.55	2.50	2.45	**0.98**	1.39	1.97	1.15	1.23	3.52	4.99	8.82	14.47
	[1,3300]	1.94	1.35	1.50	1.93	2.10	**0.88**	1.21	1.52	1.10	1.03	2.42	3.18	6.41	12.08
		average	1.32	2.40	4.29	6.61	1.80	1.25	1.87	2.09	**1.01**	3.82	7.71	10.12	20.63
600	[1,600]	0.33	**0.00**	0.08	**0.00**	0.37	0.07	**0.00**	**0.00**	0.07	**0.00**	**0.00**	0.23	0.06	3.57
	[1,900]	0.49	**0.00**	0.25	0.01	1.74	0.17	**0.00**	**0.00**	0.20	**0.00**	**0.00**	0.53	0.15	10.27
	[1,1200]	0.67	**0.00**	0.69	0.16	5.78	0.36	**0.00**	**0.00**	0.61	**0.00**	**0.00**	3.64	0.75	19.49
	[1,1500]	0.83	0.13	1.89	4.32	11.19	1.36	0.11	0.21	1.51	**0.06**	1.40	10.98	12.13	28.88
	[1,1800]	0.99	1.11	5.42	9.99	17.53	4.56	1.05	2.38	4.91	**0.67**	8.87	18.49	22.69	37.23
	[1,2100]	1.17	1.48	3.70	8.40	13.27	2.80	1.33	2.31	3.06	**1.17**	8.34	14.66	19.40	31.82
	[1,2400]	1.34	1.06	1.99	4.52	7.67	1.44	1.02	1.75	1.71	**0.96**	5.79	9.93	14.31	24.55
	[1,2700]	1.50	0.85	1.64	2.59	5.43	1.23	0.93	1.53	1.41	**0.79**	3.77	6.63	10.81	20.13
	[1,3000]	1.65	0.92	1.23	1.66	3.29	**0.76**	0.96	1.61	0.85	0.80	3.15	5.16	8.46	16.69
	[1,3300]	1.83	0.77	0.96	1.14	2.15	**0.51**	0.62	1.08	0.58	0.57	2.26	3.46	6.13	13.61
		average	0.63	1.79	3.28	6.84	1.32	0.60	1.09	1.49	**0.50**	3.36	7.37	9.49	20.62

The computation times are the same for all the algorithms. Table 8 shows the average computation times over all the processing time intervals (computation times were almost the same for all the processing time intervals). The computation times of the algorithms increase with the number of jobs. A smaller impact on the computation time has the number of machines and the number of resources. The computation time increases slightly with the number of machines and tends to decrease when the number of resources grows.

Table 8. Computation times

n	m	$l = 1$	$l = m$
100	2	0.08	0.08
300		0.46	0.33
600		3.03	1.77
100	4	0.09	0.13
300		0.59	0.57
600		3.46	2.68
100	6	0.12	0.17
300		0.77	0.57
600		4.74	4.43

Significant improvement in the quality of the results for the problems where the stages in the flowshop are balanced is a very important achievement of the proposed algorithms.

5. Conclusions

In this paper we considered the problem of preemptive scheduling in multiprocessor two-stage flowshop, subject to additional renewable resource constraints. We proposed heuristic algorithms for solving this problem. The algorithms have been tested as to their effectiveness in finding a minimum makespan schedule and their computation times. The results indicate that the proposed algorithms can produce good quality solutions in a short computation time.

Acknowledgements

We wish to thank the anonymous referees for their helpful comments and suggestions which have improved the quality of the paper.

References

Brah, S.A. (1988) Scheduling in a flow shop with multiple processors. Unpublished Ph.D. Dissertation, University of Houston, Houston, TX.

Brah, S.A. and Loo, L.L. (1999) Heuristics for scheduling in a flow shop with multiple processors. *EJOR* **113**, 113-112.

Chen, B. (1995) Analysis of classes of heuristics for scheduling a two-stage flow shop with parallel machines at one stage. *Journal of Operational Research Society* **46**, 234-244.

De Werra, D. (1984) Preemptive scheduling, linear programming and network flows. *SIAM Journal on Algebraic and Discrete Methods* **5**, 11-20.

Figielska, E. (1999) Preemptive scheduling with changeovers: using column generation technique and genetic algorithm. *Computers and Industrial Engineering* **37**, 63-66.

Figielska, E. (2005) A simulated annealing based approach to the parallel machine scheduling problem with jobs and setups using additional renewable resources. *Proc. 11th IEEE Int. Conf. Methods and Models in Automation and Robotics*, 1061-1066.

Figielska, E. (2006a) A genetic algorithm based heuristic for the parallel machine scheduling problem with setups and additional renewable resources. In: A. Cader et al., eds., *Artificial Intelligence and Soft Computing.* Academic Publishing House EXIT, Warszawa, 189-195.

Figielska, E. (2006b) A heuristic algorithm for scheduling in a two-stage multiprocessor flowshop with 0-1 resource requirements. *Proc. of the 12th IEEE International Conference on Methods and Models in Automation and Robotics*, Międzyzdroje.

Figielska, E. (2006c) Solving a flowshop scheduling problem with additional resource constraints. *Prace naukowe PW, Elektronika* **156**, 139-146.

Grabowski, J. and Janiak, A. (1984) Sequencing problem with resource constraints. In: R. Trappl, ed., *Cybernetics and System Research* **2**, Elsevier Science Publishers, Amsterdam, 329-333.

Gupta, J.N.D. (1988) Two stage hybrid flowshop scheduling problem. *Journal of Operational Research Society* **39**, 359-364.

Haouari, M. and M'Hallah, R. (1997) Heuristic algorithms for the two-stage hybrid flowshop problem. *Operations Research Letters* **21**, 43-53.

Hoogeveen, J.A., Lenstra, J.K. and Veltman, B. (1996) Preemptive scheduling in a two-stage multiprocessor flow shop is NP-hard. *EJOR* **89**, 172-175.

Janiak, A. (1986) Flow-shop scheduling with controllable operation processing times. In: *Large Scale Systems: Theory and Applications.* Pergamon Press, Oxford, 602-605.

Janiak, A. (1988a) General flow-shop scheduling with resource constraints. *International Journal of Production Research* **26**, 1089-1103.

Janiak, A. (1988b) Minimization of the total resource consumption in permutation flow-shop sequencing subject to a given makespan. *Modelling, Measurement and Control* (AMSE Press) **13**, 1-11.

Janiak, A. (1989) Minimization of the total resource consumption under a given deadline in two processor flow-shop scheduling problem. *Information*

Processing Letters **32**, 101-112.

Janiak, A. (1991) Scheduling and resource allocation problems in some flow type manufacturing process. In: G. Fandel and G. Zaepfel, eds., *Modern Production Concepts*. Springer-Verlag, Berlin, 404-415.

Janiak, A. and Portmann, M.C. (1998) Genetic algorithm for the permutation flow-shop scheduling problem with linear model of operations. *Annals of Operations Research* **83**, 95-114.

Janiak, A. (1998) Minimization of the makespan in a two-machine problem under given resource constraints. *EJOR* **107**, 325-337.

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

Kellerer, H. and Strusevich, V.A. (2002) Scheduling parallel dedicated machines under a single non-shared resource. *EJOR* **147**, 345-364.

Lawler, E.L. and Labetoulle, J. (1978) On preemptive scheduling on unrelated parallel processors by linear programming. *J. Assoc. Comput. Mach.* **25**, 612-619.

Linn, R. and Zhang, W. (1999) Hybrid flow shop scheduling: a survey. *Computers and Industrial Engineering* **37**, 57-61.

Ruiz, R. and Maroto, C. (2006) A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility. *EJOR* **169**, 781-800.

Salvador, M.S. (1973) A solution of a special class of flowshop scheduling problems. In: S.E. Elmaghraby, ed., *Symposium of the Theory of Scheduling and Its Application*. Springer-Verlag, New York, 83-91.

Serafini, P. (1996) Scheduling jobs on several machines with the job splitting property. *Operations Research* **44**, 617-628.

Słowiński, R. (1980) Two approaches to problems of resource allocation among project activities – A comparative study. *Journal of Operational Research Society* **31**, 711-723.

Słowiński, R. (1981) L'ordonnancement des taches preemptives sur les processeurs independants en presence de ressources supplementaires. *RAIRO Inform./Comp. Science*, **15**, 155-166.

Słowiński, R. and Węglarz, J. (1977) Time-minimal network model with various activity performing modes (in Polish). *Przegląd Statystyczny* **24**, 409-416.

Suresh, V. (1997) A note on scheduling of two-stage flow shop with multiple processors. *International Journal of Production Economics* **49**, 77-82.