ISSN 1330-3651
UDC/UDK 658.514:004.421.2

FLOW SHOP SCHEDULING ALGORITHM TO MINIMIZE COMPLETION TIME FOR *n*-JOBS *m*-MACHINES PROBLEM

Vladimír Modrák, R. Sudhakara Pandian

Original scientific paper

In multi stage job problems, simple priority dispatching rules such as shortest processing time (SPT) and earliest due date (EDD) can be used to obtain solutions of minimum total processing time, but may not sometimes give sequences as expected that are close to optimal. The Johnson's algorithm is especially popular among analytical approaches that are used for solving *n*-jobs, 2-machines sequence problem. In this paper the presented algorithm is based on converting an *m*-machine problem to a 2-machine problem. Based on testing and comparison with other relevant methods, the proposed algorithm is offered as a competitive alternative for practical application when solving *n*-jobs and *m*-machines problems.

Keywords: *CDS heuristics, flow shop, genetic algorithm, make-span, slope index*

Algoritam planiranja operacija "flow shop" u cilju smanjivanja vremena izvršenja kod problema *n*-poslova i *m*-strojeva

Izvorni znanstveni članak

U problemima posla s više faza, mogu se koristiti jednostavna prioritetna dispečerska pravila kao što su najkraće vrijeme obrade (PT) i najraniji datum dospijeća (EDD) za dobivanje rješenja najmanjega ukupnog vremena obrade. Međutim, ona ponekad ne daju slijed za koji se očekuje da je blizu optimalnom. Johnsonov algoritam je posebno popularan među analitičkim pristupima koji se koriste za rješavanje problema slijeda *n*-poslova i 2-stroja. Algoritam prikazan u ovom radu se temelji na pretvaranju problema *m*-strojeva u problem 2-stroja. Na temelju ispitivanja i usporedbe s drugim relevantnim metodama, predloženi algoritam se nudi kao konkurentna alternativa za praktičnu primjenu pri rješavanju problema *n*-poslova i m-strojeva.

Ključne riječi: *CDS heuristika, flow shop, genetski algoritam, indikator prioriteta, vrijeme izvršenja posla*

1 Introduction
Uvod

In a shop floor of the industry, the routings which are based upon the jobs that need to be processed on different machines are one among the major activities and therefore the resource requirements are not based upon the quantity as in a flow shop but rather the routings for the products being produced. However, both job shop and flow shop production cope with a scheduling problem to find a feasible sequence of jobs on given machines with the objective of minimising some function of the job completion times. Job completion time (make-span) can be defined as the time span from material availability at the first processing operation to the completion at the last operation [1]. Johnson [2] has shown that, in a 2-machines flow shop, an optimal sequence can be constructed. It was demonstrated later that *m*-machine flow shop scheduling problem (FSSP) is strongly NP-hard for $m \geq 3$ [3]. FSSPs can be divided into two main categories: dynamic and static. The dynamic flow shop considered is one where jobs arrive continuously over time. The static flow shop-sequencing and scheduling problem denotes the problem of determining the best sequence of jobs on each machine in the flow shop. The criterion of optimality in a flow shop sequencing problem is usually specified as minimization of make-span that is defined as the total time to ensure that all jobs are completed on all machines. If there are no release times for the jobs then the total completion time equals the total flow time. In some cases for calculating the completion times specific constraints are assumed. For example, such a situation in the FSSP arises when no idle time is allowed at machines. This constraint creates an important practical situation that arises when expensive machinery is employed [4]. The general scheduling problem for a classical shop flow gives rise to $(n!)^m$ possible schedules. With the aim to reduce the number of possible schedules it is reasonable to assume that all machines process jobs in the same order [5]. The deterministic job shop scheduling problem consists of a finite set J of n jobs to be processed on a finite set M of m machines. Each job, J_i, must be processed on every machine in its routing consisting of mi operations O_{i1}, O_{i2}..., O_{im} performed in order.

The proposed algorithm for minimizing completion time is determined for a classical static and deterministic permutation flow shop scheduling problem (PFSSP) with n jobs and m machines which is viewed as sequence problem. In the classical flow-shop sequencing and scheduling problem, queues of jobs are allowed at any of m machines in processing sequence based on assumption that jobs may wait on or between the machines [6]. In this study, the objective function for the PFSS problem corresponds to the minimization of the make-span when idle time is allowed on machines.

2 Literature review
Pregled literature

The flow-shop problem with make-span (c_{max}) criterion can be denoted as either $n/m/F/c_{max}$ or $F//c_{max}$, where both are related to an n-jobs and m-machines problem. This notation was firstly suggested by Conway et al. [7] and until now is handy. Pinedo [8] introduced the term Permutation Flow-shop Problem (PFSP) in which the processing sequence on the first machine is maintained throughout the remaining machines. Accordingly, the make-span criterion is denoted as $F/prmu/C_{max}$. Solution methods for flow shop scheduling range from heuristics developed by Palmer [9], Campbell et al. [10], and Dannenbring [11] to more complex techniques such as branch and bound [12], tabu search [13, 14, 15], genetic algorithms [16, 17] shifting bottleneck procedure [18], and ant colony algorithm [19].

The concept of a slope index in prioritizing jobs was first introduced by Page [20]. Later on Palmer [9] adopted this idea and proposed the slope index to be utilized for job sequencing in the *m*-machine flow shop problems. A simple heuristic extension of Johnson's rule to *m*-machines flow shop problem was proposed by Campbell et al [10]. This extension is known in the literature as the Campbell, Dudek, and Smith (CDS) heuristic. Its principle relies on constructing at most (*m*-1) different sequences from which the best sequence is chosen. Each sequence corresponds to the application of Johnson's rule on a new 2-machines problem. CDS heuristics will also be used in this study to compare solutions of the same PFSPs with the proposed algorithm.

Another approach to obtain minimum idle time based on the optimization of idle time at the last machine is presented in [21]. Nawaz et al. [22] proposed that a job with longer total processing time should have higher priority in the sequence. More complex heuristics was applied by Ogbu et al. [23] by using simulated annealing and by Taillard [24] by applying tabu search algorithm for make-span minimization. Nagar et al. [25] proposed a combined branch-and-bound and genetic algorithm based procedure for a flow shop scheduling problem with objectives of mean flow time and make-span minimization. Similarly, Neppalli et al. [26] used genetic algorithms in their approach to solve the 2-machine flow shop problem with the objective of minimizing make-span and total flow time. An atypical method based on an artificial immune system approach that was inspired by vertebrate immune system was presented by Engin and Doyen [27]. They used the proposed method for solving the hybrid flow shop scheduling problem with minimizing maximum completion times.

Even though the various studies suggested many approaches, it is difficult to find the simplest approach to find an optimal sequence for solving the *n*-jobs and *m*-machines flow shop scheduling problem.

In the future, scheduling approaches in this manufacturing area will need to take also market developments into consideration, especially the new manufacturing technology and advanced production control systems that will constrain the overall structure of the flow-shop manufacturing operations [28]. Keeping this in mind, scheduling algorithms to minimize make-span for *n*-Jobs *m*-Machines Problem with simplest steps will be always needful.

3
The proposed approach to multi stage flow shop sequencing
Predloženi pristup za slijed operacija u višefaznom "flow shop"

In the multi stage sequencing problem, the following assumptions are made.

- There are *n* number of jobs (J) and *m* number of machines (M).
- The order of sequence of operations in all machines is the same.
- The setup time is not considered for calculating make-span time.

The proposed approach works with simple steps as given in section 3.1. The optimum sequence is found out in step 7 that adopts the method of Johnson's algorithm [2], which is used to find out minimum make-span while 2-machine production schedules are included. The step by step algorithm is given in section 3.1

3.1
The algorithm description
Opis algoritma

Step1. Find out the sum of processing time of *n* jobs in machine M_1.

Repeat Step 1 for machines j=1, 2, 3, …, m.

Step 2. Make two groups from *m* machines in such a way that

$$\sum_{j=1}^{x} T_i \sim \sum_{j=x+1}^{m} T_i \rightarrow \text{minimum}. \quad (1)$$

Step 3. Find out the total number of machines in each group.

Let the number of machines in Group I = *a*, and the number of machines in Group II = *b*.

Step 4. Calculate total operational time *T* of jobs in each group using the formula:

a) for the Group I and Job (J_1)

$T_{J1}^{I} = (a \cdot t_{11}) + [(a-1) \cdot t_{12}] + [(a-2) \cdot t_{13}] + ... + (1 \cdot t_{1a})$.

Similarly calculate these values for jobs J2, J3, J*n*.

b) for the Group II and Job (J_1)

$T_{J1}^{II} = (b \cdot t_{1m}) + [(b-1) \cdot t_{1m-1}] + [(b-2) \cdot t_{1m-2}] + ... + (1 \cdot t_{1a+1})$.

Similarly calculate these values for jobs J2, J3, J*n*.

Step 5. Tabulate these values in two rows.

Step 6. Apply final step of Johnson's rule to find out the best sequence.

Step 7. Calculate the make-span time for the sequence obtained in step 6.

Step 8. Store the results.

3.2
The algorithm illustration
Ilustracija algoritma

To evaluate the proposed algorithm the following 6-jobs and 5-machines problem from the real life has been used. Input values for the calculation of total operational time *T* of jobs in each group are shown in Tab. 1.

Table 1 *Illustration for the problem of size 6 machines × 5 jobs*
Tablica 1. *Ilustracija za problem veličine 6 strojeva × 5 poslova*

*j**i*	J1	J2	J3	J4	J5
M1	1	1,5	1,5	1	1
M2	0,5	0,75	0,75	0,5	0,5
M3	0,5	1	0,5	0,5	0,5
M4	0,5	1	0,5	0,5	0,5
M5	0,1	0,5	0,2	0,1	0,1
M6	0,2	0,3	0,3	0,1	0,1

In the above table, each row represents machine *j* and each column represents job *i*. The processing time of an operation of the jobs is mentioned in each cell and denoted as t_{ij}.

The sum of the processing time of all 5 jobs in each machine is calculated in column T_i as shown in Tab. 2 and

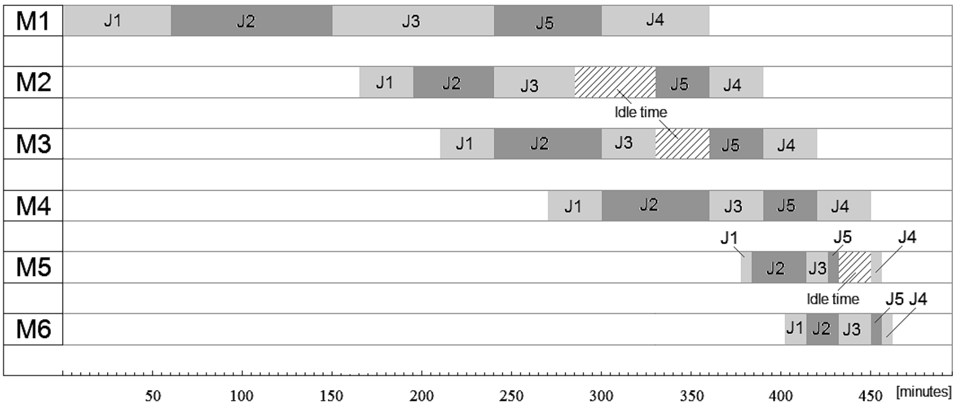

Figure 1 *Gant chart for the criteria of minimum make-span and minimum process interruptions*
Slika 1. *Gantov dijagram za kriterije minimalnog vremena izvršenja i minimum prekida procesa*

Tab. 3. Two groups are formed based on the formula as given below.

$$\sum_{j=1}^{a} T_i \sim \sum_{j=a+1}^{m} T_i \rightarrow \text{minimum} \tag{2}$$

(a = the arbitrary value from 1 to 5)

$$\sum_{j=1}^{2} T_i - \sum_{j=3}^{6} T_i = 9 - 8 \tag{3}$$

Thus, the total number of machines in each group is identified.

The number of machines in Group I (Tab. 2), a = 2 (M1 and M2 are in Group-I, noted as I).

The number of machines in Group II (Tab. 3), b = 4 (M3, M4, M5 and M6 are in Group-II, noted as II).

Table 2 *Group I consisting of two machines*
Tablica 2. *I. skupina koja se sastoji od dva stroja*

$j \backslash i$	J1	J2	J3	J4	J5	T_i	ΣT_i
M1	1	1,5	1,5	1	1	6	9
M2	0,5	0,75	0,75	0,5	0,5	3	

Table 3 *Group II consisting of four machines*
Tablica 3. *II. skupina koja se sastoji od četiri stroja*

$j \backslash i$	J1	J2	J3	J4	J5	T_i	ΣT_i
M3	0,5	1	0,5	0,5	0,5	3	8
M4	0,5	1	0,5	0,5	0,5	3	
M5	0,1	0,5	0,2	0,1	0,1	1	
M6	0,2	0,3	0,3	0,1	0,1	1	

Subsequently, for the identified groups I and II the values of T^{I}_{Ji} and T^{II}_{Ji} (for i=1 to n) are calculated for all five jobs.

$T^{I}_{J1} = (2\times1,0) + 0,50 = 2,5$
$T^{I}_{J2} = (2\times1,5) + 0,75 = 3,75$
$T^{I}_{J3} = (2\times1,5) + 0,75 = 3,75$
$T^{I}_{J4} = (2\times1,0) + 0,50 = 2,5$
$T^{I}_{J5} = (2\times1,0) + 0,50 = 2,5$
$T^{II}_{J1} = (4\times0,2) + (3\times0,1) + (2\times0,5) + 0,5 = 2,6$
$T^{II}_{J2} = (4\times0,3) + (3\times0,5) + (2\times1,0) + 1,0 = 5,7$
$T^{II}_{J3} = (4\times0,3) + (3\times0,2) + (2\times0,5) + 0,5 = 3,3$
$T^{II}_{J4} = (4\times0,1) + (3\times0,1) + (2\times0,5) + 0,5 = 2,2$
$T^{II}_{J5} = (4\times0,1) + (3\times0,1) + (2\times0,5) + 0,5 = 2,2$

The T^{I}_{Ji} and T^{II}_{Ji} values are tabulated as shown in Tab. 4.

Table 4 *The sum of values of two groups*
Tablica 4. *Zbroj vrijednosti dviju skupina*

Groups\jobs	J1	J2	J3	J4	J5
T^{I}_{J}	2,5	3,75	3,75	2,5	2,5
T^{II}_{J}	2,6	5,7	3,3	2,2	2,2

As per the step 6 of the algorithm, the best sequences obtained in this method are J1-J2-J3-J5-J4 (or) J1-J2-J3-J4-J5.

The make-span, when idle time is allowed on machines, is calculated for the J1-J2-J3-J5-J4 sequence (see Tab.5) since both sequences in the given case bring identical scheduling results.

Table 5 *Proposed method J1-J2-J3-J5-J4*
Tablica 5. *Predložena metoda J1-J2-J3-J5-J4*

j	M1		M2		M3		M4		M5		M6	
i	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out
J1	0	1	1	1,5	1,5	2	2	2,5	2,5	2,6	2,6	2,8
J2	1	2,5	2,5	3,25	3,25	4,25	4,25	5,25	5,25	5,75	5,75	6,05
J3	2,5	4	4	4,75	4,75	5,25	5,25	5,75	5,75	5,95	6,05	6,35
J5	4	5	5	5,5	5,5	6	6	6,5	6,5	6,6	6,6	6,7
J4	5	6	6	6,5	6,5	7	7	7,5	7,5	7,6	7,6	7,7

With the aim to combine the criterion for calculating the minimum make-span schedules when idle time is allowed on machines along with the criterion for minimum process interruptions it is possible to create job schedules by the manner shown in Gant chart in Fig. 1.

4 Comparison with Benchmark Algorithms
Usporedba s repernim algoritmima

To compare the proposed algorithm with the benchmark algorithms, the next three distinct algorithms are used: CDS heuristics, Slope index method and Genetic Algorithm. The make-spans for CDS method and Slope algorithm are also calculated and displayed in Tab. 6 and Tab. 7. The sequence obtained by using GA for the same PFSS problem equals the sequence calculated by the proposed method. Moreover these four methods have been employed for finding the best sequence with the other four problems to achieve more reliable results. For this purpose we selected flow shop problems, which are shown in Fig. 2 (a-d). The results obtained with the benchmark methods are compared and shown in Tab. 8.

4.1 CDS heuristics
CDS heuristika

As outlined above, the CDS heuristics algorithm [9] is basically an extension of the Johnson's algorithm. The focus of the heuristic is the minimization of make-span in a deterministic flow shop problem. The CDS heuristic forms in a simple manner a set of an *m*-1 artificial 2-machine sub-problem for the original *m*-machine problem by summing the processing times in a manner that combines M1, M2,..., M*m*-1 to pseudo machine 1 and M2, M3,... M*m* to pseudo machine 2. Finally, each of the 2-machine sub-problems is then solved using the Johnson's 2-machines algorithm. The best of the sequence is selected as the solution to the original *m*-machine problem.

For the given flow shop problem of size 6×5 as given in Tab. 1, using this heuristic the J2-J3-J1-J5-J4 sequence has been calculated and the make-span calculation is displayed in Tab. 6.

*j**i*	J1	J2	J3	J4
M1	24	61	22	21
M2	7	9	8	6
M3	7	5	6	8
M4	29	15	14	32

(a)

*j**i*	J1	J2	J3	J4	J5	J6	J7
M1	3	2	4	5	1	3	5
M2	5	5	8	7	2	5	2
M3	7	8	1	6	8	4	8
M4	1	1	6	1	4	6	4
M5	6	6	7	8	6	8	6
M6	9	7	9	4	7	1	3
M7	4	9	1	3	4	2	2

(b)

*j**i*	J1	J2	J3	J4
M1	7	6	5	8
M2	5	6	4	3
M3	2	4	5	3
M4	3	5	6	2
M5	9	10	8	6

(c)

*j**i*	J1	J2	J3	J4	J5	J6	J7
M1	5	2	4	2	5	2	1
M2	1	1	5	1	4	8	5
M3	4	2	2	5	8	5	4
M4	5	5	3	4	9	5	2
M5	8	4	5	6	2	6	4
M6	2	5	8	3	4	5	5
M7	4	3	2	2	8	8	2
M8	8	2	5	2	5	5	8

(d)

Figure 2 *Input data set for testing flow shop problems (operational time in hours)*
Slika 2. *Niz ulaznih podataka za ispitivanje problema protočne radionice (operativno vrijeme u satima)*

Table 6 *Make-span calculation for the J2-J3-J1-J5-J4 sequence*
Tablica 6. *Izračun vremena izvršenja za slijed J2-J3-J1-J5-J4*

j	M1		M2		M3		M4		M5		M6	
i	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out
J2	0	1,5	1,5	2,25	2,25	3,25	3,25	4,25	4,25	4,75	4,75	5,05
J3	1,5	3	3	3,75	3,75	4,25	4,25	4,75	4,75	4,77	5,05	5,35
J1	3	4	4	4,5	4,5	5	5	5,5	5,5	5,6	5,6	5,8
J5	4	5	5	5,5	5,5	6	6	6,5	6,5	6,6	6,6	6,7
J4	5	6	6	6,5	6,5	7	7	7,5	7,5	7,6	7,6	7,7

4.2 Slope index method
Metoda indikatora prioriteta

A heuristic has been developed by Palmer [9] in an effort to use Johnson's rule for $m \geq 3$, since for $m=2$, this algorithm is slightly different from Johnson's algorithm. The idea of this procedure is to give priority to some jobs so that the jobs with the processing times that tend to increase from machine to machine will receive higher priority, while the jobs with the processing times that tend to decrease from machine to machine will receive lower priority.

The slope index (*SI*) for job *i* is calculated as:

$$SI_i = \sum_{j=1}^{m} (2j - m - 1) t_{ij}, i = 1, 2, ...n. \quad (4)$$

Then a permutation sequence is determined by ordering the jobs in no increasing order of SI_i such as:

$$SI_{i1} \geq SI_{i2} \geq .... \geq SI_{in}. \quad (5)$$

For the original flow shop 5-jobs and 6-machines problem as given in Tab. 1, using this heuristic the J1-J4-J5-J2-J3 sequence has been calculated and the make-span calculation is displayed in Tab. 7.

Table 7 *Slope index method J1-J4-J5-J2-J3*
Tablica 7. *Metoda indikatora prioriteta J1-J4-J5-J2-J3*

j	M1		M2		M3		M4		M5		M6	
i	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out
J1	0	1	1	1,5	1,5	2	2	2,5	2,5	2,6	2,6	2,8
J4	1	1,5	1,5	2	2	2,5	2,5	3	3	3,1	3,1	3,2
J5	1,5	2,5	2,5	3	3	3,5	3,5	4	4	4,1	4,1	4,2
J2	2,5	4	4	4,8	4,8	5,8	5,8	6,8	6,8	7,3	7,3	7,6
J3	4	5,5	5,5	6,3	6,3	6,8	6,8	7,3	7,3	7,5	7,6	7,9

4.3 Genetic Algorithm (GA)
Genetski algoritam (GA)

The Genetic Algorithm is a probabilistic approach which deals with probability [16]. In this paper GA is used to search for optimal solution and the results obtained from the GA are compared with the results of the proposed algorithm as shown in Tab. 8. From this table it is inferred that the proposed approach gives as good results as those of the well known meta-heuristics, GA [17]. This shows the ability of the proposed algorithm with simpler steps. The following steps shown in Fig. 3 provide the pseudo code of GA used for this purpose.

Step 1 Generate initial population with job sequence as solution strings.
Step 2 Find out the fitness value (makespan value)
Step 3 Reproduction of the strings with better makespan values
Step 4 Apply cross over with crossover probability P_c=0,5
Step 5 Apply mutation with mutation probability P_m = 0,1
Step 6 Find out the fitness value.
Step 7 Store the best values.
Step 8 Go to step 3 and iterate till the generation value (gen=500).

Figure 3 *The pseudo code of GA*
Slika 3. *Pseudo kod genetskog algoritma*

5 Discussion and conclusion
Rasprava i zaključak

The present study deals with sequence-dependent operations, the sequencing problem which is quite common in many industries. The main idea is to minimize the make-span time thus reducing the idle time of both jobs and machines since these criteria are often applied for operational decision-making in scheduling. Based on the tested problems it can be concluded that the proposed approach produces results comparable with the benchmark algorithms as shown in Tab. 8.

Table 8 *Comparative results of make-span*
Tablica 8. *Usporedni rezultati vremena izvršenja*

S. No	Number of Machines	Number of Parts	Make-span (in hours)			
			CDS Algorithm	Slope Index	GA	Proposed Approach
1	4	4	156,0	157,0	156,0	156,0
2	5	4	51,0	51,0	51,0	51,0
3	6	5	7,7	7,9	7,7	7,7
4	7	7	6,7	7,5	6,7	6,7
5	8	7	7,1	6,9	6,7	6,7

Many heuristics and meta-heuristics can find quick, feasible solutions to such sequencing problems that involve multiple jobs and machines and sequence-dependent operations. But, as far as simplicity of the algorithm and promising results are concerned, the proposed method is more effective than the existing methods. In realistic situation, the proposed algorithm can be used such as it is without any modification and come out with acceptable results. In that manner the approach can be recommended for industries that deal with variety of parts and machines with more operations.

Acknowledgement
Zahvala
This research has been partially supported by the financing from VEGA grant scheme No. 1/4153/07.

6 References
Literatura

[1] Modrák, V.; Modrák, J. Relationships between Batch Sizes, Sequencing and Lead-times. Proceedings of the 6th International Conference on Informatics in Control, Automation and Robotics, Intelligent Control Systems and Optimization, Milan, Italy, July 2-5, 2009. INSTICC Press 2009, 380-383.
[2] Johnson, S. M. Optimal Two and Three Stage Production Schedules with Set-Up Times, Included, Naval Research Logistics Quarterly, 1(1954), 61-68.
[3] Garey, M. R. D.; Johnson, D. S.; Sethi, R. The complexity of flow shop and job shop scheduling. Mathematics of Operations Research, 1(1976), 117–129.
[4] Ruiz, R. Vallada, E.; Fernandez-Martinez, C. In. Computational Intelligence in Flow Shop and Job Shop Scheduling, Volume 230/2009, Springer Berlin / Heidelberg. (2009)
[5] Gupta, J. N. D. Analysis of combinatorial approach to flow shop scheduling problems. Journal of the Operational Research Society, 26(1975), 431-440.
[6] Allahverdi, A.; Gupta, J. N. D.; Aldowaisan, T. A review of scheduling research involving set-up considerations. Omega, 27(1999), 219–239.
[7] Conway, R. W.; Maxwell, W. L.; Miller, L. W. Theory of Scheduling, Addison-Wesley: Reading, MA, 1967.
[8] Pinedo, M. Scheduling: Theory, Algorithms and Systems. Prentice Hall, New Jersey, second edition, 2002.
[9] Palmer, D. S. Sequencing jobs through a multi-stage process in the minimum total time – a quick method of obtaining a near optimum, Operations Research. Q. 16(1965), 101-107.
[10] Campbell, H. G.; Dudek, R. A.; Smith, M. L. A Heuristic Algorithm for the n-Job, m-Machine Sequencing Problem, Management Science, 16, 10(1970), 630-637.
[11] Dannenbring, David G. An Evaluation of Flow Shop Sequencing Heuristics, Management Science, 23, 11(1977), 1174-1182.
[12] Brucker, P.; Jurisch, B.; Sievers, B. A branch and bound algorithm for the job shop scheduling problem. Discrete Applied Mathematics, 49, 1(1994), 109–127.
[13] Gendreau, M.; Laporte, G.; Semet, F. A tabu search heuristic for the undirected selective travelling salesman problem, European Journal of Operational Research, Elsevier, 106, 2-3(1998), 539-545.
[14] Nowicki, E.; Smutnicki, C. A fast taboo search algorithm for the job shop problem. Management Science, 42 ,6(1996), 797–813.
[15] Logendran, R.; de Szoeke, P.; Barnard, F. Sequence-dependent group scheduling problems in flexible flow shops. International Journal of Production Economics, 102 (2006), 66–86.
[16] Manikas. A.; Chang, Y. L. Multi-criteria sequence-dependent job shop scheduling using genetic algorithms Computers & Industrial Engineering, 56 (2009), 179–185.
[17] Murata, T.; Ishibuchi. H.; Tanaka, H. Genetic Algorithms for Flow shop Scheduling Problems, Computers & Industrial Engineering, 30, 4 (1996), pp. 1061-1071.
[18] Balas, E. and A. Vazacopoulos. Guided Local Search with Shifting Bottleneck for Job Shop Scheduling. Management Science, 44, 2(1998), 262-275.
[19] Blum, C.; Sampels, M. An Ant Colony Optimization Algorithm for Shop Scheduling Problems. Journal of Mathematical Modelling and Algorithms, 3, 3(2004), 285-308.
[20] Page, E. S. An Approach to Scheduling of Jobs on the Machines, J. Royal Stat. Soc., V 23, (1961), pp. 484-492.
[21] Gupta, J. N. D.: Heuristic algorithms for multistage flow shop scheduling problem, AIIE Transactions, 4, (1)(1972), 11-18.
[22] Nawaz. M.; Enscore, E.; Ham, I. A heuristic algorithm for the m machine, n job flow shop sequence problem, OMEGA, 11, 1(1983), 91-95.
[23] Ogbu, F. A.; Smith, D. K. The application of the simulated annealing algorithm to the solution of the n/m/Cmax flow shop problem, Computers & Operations Research, 17, 3(1990), 243-253.
[24] Taillard, E. Some efficient heuristic methods for the flow shop sequencing problem, European Journal of Operational Research, 47, 1(1990), 65-74.
[25] Nagar, A.; Heragu, S. S.; Haddock, J. A combined branch-and-bound and genetic algorithm based approach for a flow shop-scheduling problem. Annal. Oper. Res., 63(1996), 397–414.
[26] Neppalli, V. R.; Chen, C. L.; Gupta, J. N. D. Genetic algorithms for the two-stage bicriteria flow shop problem. Eur. J. Oper. Res., 95(1996), 356–373.
[27] Engin, O.; Doyen, A. A new approach to solve hybrid flow shop scheduling problems by artificial immune system. Future Generation Computer Systems, 20(2004), 1083–1095.
[28] Valíček, J.; Hloch, S.; Kozak, D. Surface geometric parameters proposal for the advanced control of abrasive waterjet technology. The International Journal of Advanced Manufacturing Technology. 41, 3-4 (2009), 323-328.

Authors' addresses
Adrese autora

Vladimír Modrák
Faculty of Manufacturing Technologies
Technical University of Košice
Bayerova 1, Prešov, Slovakia
vladimir.modrak@tuke.sk

R. Sudhakara Pandian
Department of Mechanical Engineering
Kalasalingam University
Krishnankoil-626190, Virudhunagar, India
sudhakarme@rediffmail.com