REVIEW ON SOLVING THE JOB SHOP SCHEDULING PROBLEM: RECENT DEVELOPMENT AND TRENDS

Ivan Lazár, M.Sc.,
Faculty of Manufacturing Technologies
Technical University of Kosice with a seat in Presov,
Bayerova 1, 080 01 Prešov,
Slovak Republic
ivan.lazar@tuke.sk

Abstract

This article discusses the development of the Job Shop problem and the development of methods to be used in solving JSSP. It also defines the groups JSS Problems, which are divided according to the complexity of the solution. The article includes the evaluation of publications research and methods research that are used in various publications.

Key words: job shop, scheduling problems, process layout

INTRODUCTION

The job-shop problem is to schedule a set of jobs on a set of machines, subject to the constraint that each machine can handle at most one job at a time and the fact that each job has a specified processing order through the machines. The objective is to schedule the jobs so as to minimize the maximum of their completion times.

This problem is not only NP-hard it is also has the well-earned reputation of being one of the most computationally stubborn combinatorial problems considered to date. This intractability is one of the reasons the problem has been so widely studied. Indeed, some of the excitement in working on the problem no doubt arose from the fact that a specific instance, with 10 machines and 10 jobs, dealt in a book by Muth and Thompson [16] remained unsolved for over 20 years. This particular instance was finally settled in 1985 by Carlier and Pinson [2], [3] that incorporates the ideas of Lageweg et. al [9].

The remainder of the paper is structured as follows: The next section outlines research methodology employed. Then, a position of the Job Shop production properties in the context of layout design is analyzed. The fourth section focuses on defining of Job Shop Scheduling Problem (JSSP). In the further section, definitions and a classification of job-shops is presented. Subsequently, a review of frequent approaches and methods for JSSP solving is treated. A final section of the paper summarizes findings of this review on mapping the field over the last 5 years.

RESEARCH METHODOLOGY

Usually quantitative research is based on a large representative survey and its outcomes are reliable data that can be generalized. As it has been mentioned above, the survey of the Job shop scheduling problem is aimed to map the field over the last 5 years and is focused:

- to quantify of research effort in developing a wide range of approaches and methods for solving JSSP
- and to determine which of these methods are most commonly used in solving this problem.

In this quantitative research were used especially scientific articles registered in Science Direct and Scirus. Keywords applied in search engines were: job shop, scheduling problems, heuristics and metaheuristics methods, genetic algorithm, tabu search, local search, linear and dynamic programming, etc.

Relevant findings compiled from the literature review are graphically shown on Figures 2 and 3 and also briefly commented in the final sections.

POSITION OF THE JOB SHOP PRODUCTION SYSTEM

According to Groover [7] there are three types of production associated with discrete-product manufacture:

1. Job shop production (low volume production),

2. Batch production (Medium- sized lots of the same item or product),

3. Mass production (two categories of mass production can be distinguished):

a) quantity production (production of simple single parts such as screws),

b) flow production (production of complex single parts such as automotive engine blocks) [7].

This classification can also serve for plants used in the process industries:

For each type of production is more or less suitable one of the four principal types of plant layout:

I) Fixed-position layout (large unites, such as a ship)

II) Process layout (according to you, Technology-oriented manufacturing)

III) Product-flow layout (according to you Object-oriented manufacturing)

Typical layouts for the given types of production are specified in Table 1.

Table 1 Possible layouts for the given types of production

Type of plant layout	Type of production
Fixed-position layout	Job shop production
Process layout	Job shop production
	Batch production
Product-flow layout	Batch production
	Mass production

Colored boxes demarcate a field of the cellular manufacturing systems. Thus manufacturing system designers have a dilemma for batch production in deciding when to apply Process layout and when to apply for the same one Product-flow layout. This issue can be solved through testing algorithm presented by Modrak [14]. Cellular manufacturing is currently a topical issue especially for manufacturers due to fact that process layout in case of batch production is not enough to pass customer requirements [15]. Accordingly, for many manufacturers, the actual issue is transformation of Process layouts to Cellular layouts as it is shown in Figure 1.

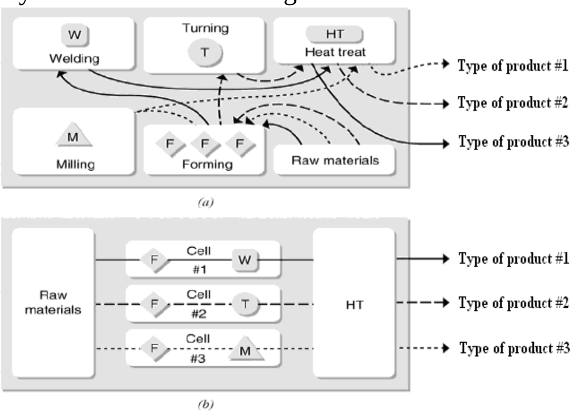

Figure 1 Transformation from Process layout (a) to Cellular layout (b)

DEFINITION AND CLASSIFICATION OF THE JOB-SHOP PROBLEM

The job-shop problem can be formulated as follows. Given are *m* machines $M_1, M_2, \ldots, M_m$ and *n* jobs $J_1, J_2, \ldots, J_n$. Job J_j consists of n_j operations O_{ij} $(i = 1, \ldots, n_j)$ which have to be processed in the order O_{1j} O_{2j}, ..., $O_{n_j j}$. It is convenient to enumerate all operations of all jobs by $k = 1, \ldots, N$ where $N = \sum_{j=1}^{n} n_j$. For each operation $k = 1, \ldots, N$ we have a processing time $pk > 0$ and a dedicated machine $M(k)$. k must be processed for p_k time units without preemptions on $M(k)$. Additionally a dummy starting operation 0 and a dummy finishing operation $N + 1$, each with zero processing time, are introduced. We assume that for two succeeding operations $k = O_{ij}$ and $s(k) = O_{i+1,j}$ of the same job $M(k) \neq M(s(k))$ holds. Let S_k be the starting time of operation k.

Then $C_k = S_k + p_k$ is the finishing time of k and (S_k) defines a schedule. A schedule (S_k) is feasible if for any succeeding operations k and $s(k)$ of the same job $S_k + p_k \leq S_{s(k)}$ holds and for two operations k and h with $M(k) = M(h)$ either $S_k + p_k \leq S_h$ or $S_h + p_h \leq S_k$. One has to find a feasible schedule (S_k) which minimizes the makespan $max_{k=1}^{N} C_k$.

Lageweg et al. in 1981 developed a computer program MSPCLASS for an automatic classification of scheduling problems. This program based on the α |β |γ-classification scheme calculates problems which are:

- <u>maximal polynomially solvable:</u> the hardest problems which are polynomially solvable, i.e. problems which are known to be polynomially solvable, but any harder cases are not known to be polynomially solvable,
- <u>maximal pseudopolynomially solvable</u>: the hardest problems which are known to be pseudopolynomially (but not polynomially) solvable,
- <u>minimal *NP*-hard</u>: the easiest problems which are *NP*-hard, i.e. problems which are known to be *NP*-hard, but any easier cases are not known to be *NP*-hard,
- <u>minimal open:</u> problems for which the complexity status is not known, but all easier cases are known to be polynomially solvable,
- <u>maximal open:</u> problems for which the complexity status is not known, but all harder cases are known to be *NP*-hard [9].

Table 2 Job-shop problems with preemption

Job-shop problems with preemption	
Maximal polynomially solvable	
$J\|prec; r_i; n = 2; pmtn\|\sum w_i U_i$	Sotskov (1991)
$J\|prec; r_i; n = 2; pmtn\|\sum w_i T_i$	Sotskov (1991)
Maximal pseudopolynomially solvable	
$J\|prec; r_i; n = k; pmtn\|\sum w_i U_i$	Middendorf & Timkovsky (1999)
$J\|prec; r_i; n = k; pmtn\|\sum w_i T_i$	Middendorf & Timkovsky (1999)
Minimal NP-hard	
$J2\|n = 3; pmtn\|C_{max}$	Brucker et al. (1999B)
$J2\|pmtn\|C_{max}$	Lenstra & Rinnooy Kan (1979)
$J2\|n = 3; pmtn\|\sum C_i$	Brucker et al. (1999B)
$J2\|pmtn\|\sum C_i$	Lenstra (-)

Table 3 Job-shop problems without preemption

Job-shop problems without preemption	
Maximal polynomially solvable	
$J2\|p_{ij}=1;r_i\|C_{max}$	Timkovsky (1997)
$J2\|p_{ij}=1\|\sum C_i$	Kubiak&Timkovsky (1996)
$J2\|p_{ij}=1\|\sum U_i$	Kravchenko (1999A)
$J\|prec;p_{ij}=1;r_i;n=k\|\sum w_iU_i$	Brucker& Kraemer (1996)
$J\|prec;r_i;n=2\|\sum w_iU_i$	Sotskov (1991)
$J2\|n=k\|\sum w_iU_i$	Brucker et al. (1997A)
$J\|prec;p_{ij}=1;r_i;n=k\|\sum w_iT_i$	Brucker& Kraemer (1996)
$J\|prec;r_i;n=2\|\sum w_iT_i$	Sotskov (1991)
$J2\|n=k\|\sum w_iT_i$	Brucker et al. (1997A)
Maximal pseudopolynomially solvable	
$J\|prec;r_i;n=k\|\sum w_iU_i$	Middendorf&Timkovsky (1999)
$J2\|p_{ij}=1\|\sum w_iU_i$	Kravchenko (1999A)
$J\|prec;r_i;n=k\|\sum w_iT_i$	Middendorf&Timkovsky (1999)
Minimal NP-hard	
$J3\|n=3\|C_{max}$	Sotskov&Shakhlevich (1995)
$J2\|\|C_{max}$	Lenstra&RinnooyKan (1979)
$J2\|chains;p_{ij}=1\|C_{max}$	Timkovsky (1985)
$J3\|p_{ij}=1\|C_{max}$	Lenstra&RinnooyKan (1979)
$J2\|p_{ij}=1;r_i\|\sum C_i$	Timkovsky (1998)
$J3\|n=3\|\sum C_i$	Sotskov&Shakhlevich (1995)
$J2\|\|\sum C_i$	Garey et al. (1976)
$J2\|chains;p_{ij}=1\|\sum C_i$	Timkovsky (1998)
$J3\|p_{ij}=1\|\sum C_i$	Lenstra (-)
$J2\|p_{ij}=1\|\sum w_iC_i$	Timkovsky (1998)
$J2\|p_{ij}=1;r_i\|\sum w_iC_i$	Timkovsky (1998)
$J2\|p_{ij}=1;r_i\|\sum U_i$	Timkovsky (1998)
$J2\|p_{ij}=1\|\sum w_iU_i$	Kravchenko (1999)
$J2\|p_{ij}=1\|\sum w_iT_i$	Timkovsky (1998)

Table 4 Job-shop problems with no-wait

Job-shop problems with no-wait	
Maximal polynomially solvable	
$J2\|p_{ij}=1;no-wait\|\sum C_i$	Kravchenko (1998)
$J\|prec;r_i;n=k;no-wait\|\sum w_iU_i$	Baptiste et al. (2004)
$J\|prec;r_i;n=k;no-wait\|\sum w_iT_i$	Baptiste et al. (2004)
Maximal pseudopolynomially solvable	
$J2\|p_{ij}=1;no-wait\|C_{max}$	Timkovsky Kubiak (1989)
Minimal NP-hard	
$J2\|p_{ij}=1;no-wait\|C_{max}$	Timkovsky Kubiak (1989)
$J2\|chains;p_{ij}=1;no-wait\|C_{max}$	Timkovsky (1998)
$J2\|no-wait\|C_{max}$	Sahni& Cho (1979)
$J3\|p_{ij}=1;no-wait\|C_{max}$	Sriskandarajah&Ladet (1986)
$J2\|p_{ij}=1;r_i;no-wait\|L_{max}$	Timkovsky (1998)
$J2\|chains;p_{ij}=1;no-wait\|\sum C_i$	Timkovsky (1998)
$J2\|no-wait\|\sum C_i$	Roeck (1984)
$J2\|p_{ij}=1;r_i;no-wait\|\sum C$	Timkovsky (1998)
$J3\|p_{ij}=1;no-wait\|\sum C_i$	Sriskandarajah&Ladet (1986)
$J2\|p_{ij}=1;no-wait\|\sum w_iC_i$	Timkovsky (1998)
$J2\|p_{ij}=1;no-wait\|\sum w_iT_i$	Timkovsky (1998)

APPROACHES AND METHODS TO SOLVE JSSP

Brucker and Schile [21] were the first authors to describe this problem in 1990. They developed a polynomial graphical algorithm for a two-job problem. In the scheduling of job-shops, the most common methodology is materials requirement planning (MRP) [1]. However, MRP is mostly a planning tool and is not really designed for detailed-level scheduling. In many companies, scheduling is performed by experienced shop-floor personnel with pencil, paper, a few graphical aids (such as Gantt chart) and perhaps a modern industrial database [6], [10]. Simple dispatching rules are often used for solving immediate problems, such as sequencing at the work-center level. The result can be scheduling chaos, where completion dates cannot be predicted and work-in-process (WIP) inventory builds [10]. Sometimes, even high-level management must chase down high-priority jobs on the shop floor. Many dispatching rules have been presented and implemented based on due dates, criticality of operations, processing times, and resource utilization. The „critical ratio“ defined by one definition as a ratio of remaining processing time over remaining time to due date, has been very popular in job-shops [6]. More complicated heuristics take into account some combination of the above factors. For example, Viviers´ algorithm incorporates three priority classes in the shortest processing tim (SPT) rule [20]. Each job is assigned an index equal to its processing time plus a value graded to its priority class. High-priority jobs have low index values and are processed first according to the SPT rule. Heuristics have been comparatively evaluated. Many artificial intelligence (AI) approaches also use dispatching rules or heuristics for scheduling [5], [8]. It is generally very difficult to evaluate the performance of schedules generated by these methods. The result may also depend upon the initial ordering of jobs. This implies that minnor changes in jobs and/or resource availability from one day to the next may result in quite different schedules. There has been a great deal of effort concentrated on optimalization methodologies.

However, exact algorithms are not effective for solving JSSP and large instances. Several heuristic procedures have been developed in recent years for the JSSP [17]. The methods in this category include dynamic programming and the branch-and-bound method, simulated annealing (SA) and genetic algorithm (GA) [4]. Because the number of possible sequences growts exponentially as the problem size, these methods become very computational intensive for even small-sized job-shops [9]. There are a plenty heuristic procedures and rules to assist in this endeavour. However, rules leading to the optimum schedule have been elusive. The predominant scheduling methods now used are specifically tailored to the type of job shop.

Generally, various rules are tried and those giving the best result are used as a starting point. The human expert sifts the schedule through his experience filter, negotiates with affected parties, and finalizes a schedule. Expert systems are beginning to impact in this area. By assuming some of the filtering and negotiating roles of the human expert, they can allow schedulers to look at more alternatives and/or produce more timely schedules.

However, the meta-heuristics methods have led to better results than the traditional dispatching or greedy heuristic algorithm. JSSP could be turned into the Job-shop scheduling problem when a routing is chosen, so when solving JSSP, hierarchical approach and integrated approach have been used [11], [12]. The hierarchical approach could reduce difficulty by decomposing the JSSP into a sequence of sub-problems. Dauzère-Pérès and Paulli [19] solved the routing subproblem using some existing dispatching rules, and then solved the scheduling sub-problem by different tabu search methods. Integrated approach could achieve better results, but it is rather difficult to be implemented in real operations. Mati, et al [13] proposed different tabu search heuristic approach to solve the JSSP using an integrated approach. Mastrolilli and Gambardella [11] proposed some neighborhood functions for the JSSP, which can be used in meta-heuristic optimization techniques, and achieve better computational results than any other heuristic developed so far, both in terms of computational time and solution quality. GA is an effective meta-heuristic to solve combinatorial optimization problems, and has been successfully adopted to solve the JSSP.

Table 5 Survey of the JSSP problem approaches

AUTHORS	YEAR	MET	SIZE
F. Pezzella, G. Morganti, G. Ciaschett	2008	GA	20x15
Jin-hui Yang, et. al.	2008	MA	10x10
Márton Drótos, et. al.	2009	N-A	N-A
Guan-Chun Luh, Chung-Huei Chueh	2009	IA	30x10
Gromicho, et.al.	2009	DP	N-A
Shi Qiang Liu, Erhan Kozan	2009	SBP	N-A
Christian Artigues, et. al	2009	B and B	N-A
Ce´sar Rego, Renato Duarte	2009	F&F	15x15
Shijin Wang, Jianbo Yu	2010	H	10x10
E. Moradi, et. al.	2010	CDR, NRGA	30x10
Shijin Wang, Jianbo Yu	2010	H	30x10
Leila Asadzadeh, Kamran Zamanifa	2010	GA	30x10
L. De Giovanni, F. Pezzella	2010	GA	20x20
Rui Zhang, Cheng Wu	2010	SAA	50x10
Deming Lei	2010	RKGA	15x10
Wannaporn Teekeng, Arit Thammano	2011	FrL, FL	20x15
Guohui Zhang, Liang Gao,Yang Shi	2011	GA	20x15
Rubiyah Yusof, et. al.	2011	MICRO GA	30x10
Veronique Sels, et. al.	2011	MH	2x20
Marnix Kammer, et. al.	2011	LS	20x20
Yazid Mati, et. al.	2011	LS	50x8
R.Tavakkoli-Moghaddam, et. al.	2011	PSO	100x20
Darrell F. Lochtefeld, Frank W. Ciarallo	2011	MOEAs	50x10
Min Liu, et. al.	2011	GA, SA	10x10
Liang Gao, et. al.	2011	MA	15x15
Mati, Yazid, et. al.	2011	LS	10x10
Ye Li , Yan Che	2011	TPA	20x10
Rui Zhang' Cheng Wu	2011	TSA	N-A
Moslehi Ghasem, Mahnam Mehdi	2011	PSO, LS	N-A
Jianchao Tang, et. al.	2011	PSO, GA	20x15

- *MET-Methods*
- *(MOEAs)-Multiple Objective Evolutionary Algorithms,*
- *(CDR)-Composite Dispatching Rule,*

- *(NSGA-II)-Non Dominated Sort Genetic Algorithm,*
- *(NRGA)-Non Ranking Genetic Algorithm,*
- *(SAA)-Simulated Annealing Algorithm,*
- *(MA)-Memetic Algorithm,*
- *(F&F)-Filter-and-Fan,*
- *(TPA)-Team Process Algorithm,*
- *(TSA)-Tree Search Algorithm,*
- *(FrL)-Frog Leaping,(FL)-Fuzzy Logic,*
- *(RKGA)-Random Key Genetic Algorithm,*
- *(SA)-Simulated Annealing,*
- *(SBP)-Shifting Bottleneck Procedure algorithm,*
- *(IA)-Immune Algorithm,*
- *(DP)-Dynamic Programming*

In the next graph in figure 2 are shown, as consistent with the second objective of this study, methods that are most commonly used in solving this problem in order of frequency. The topicality of genetic algorithm seems to be beyond a reasonable doubt due to computational results demonstrating the superiority in terms efficiency and effectiveness. As regards to memetic algorithms (MA), they represent one of the growing areas of research in evolutionary computation and are widely used as a synergy of evolutionary or any population-based approach with separate individual learning or local improvement procedures for problem search.

One of the several approaches that may be useful for the JSSP with the objective to minimize the maximum completion time is particle swarm optimization (PSO)-based memetic algorithm (MA). In the PSO-based MA algorithm both PSO-based searching operators and some special local searching operators are employed to balance the exploration and exploitation abilities. In particular, this algorithm applies the evolutionary searching mechanism of PSO, which is characterized by individual improvement, population cooperation, and competition to effectively perform exploration.

The second important group of algorithms includes well-known Local search. Iterative LS is powerful optimization procedures which have been successfully applied to a number of JSSPs.

A general description of the pertinent findings obtained from the presented survey is possible to demonstrate by the following graphs.

In accordance with the methodology section and the first objective of this study, one can say that the number of articles dealing with the JSSP began rapidly to expand starting from 2006 (see Figure 2 and 3). This can be perceived as evidence that JSSP is quite popular research topic, which is dealt by increasing number of authors.

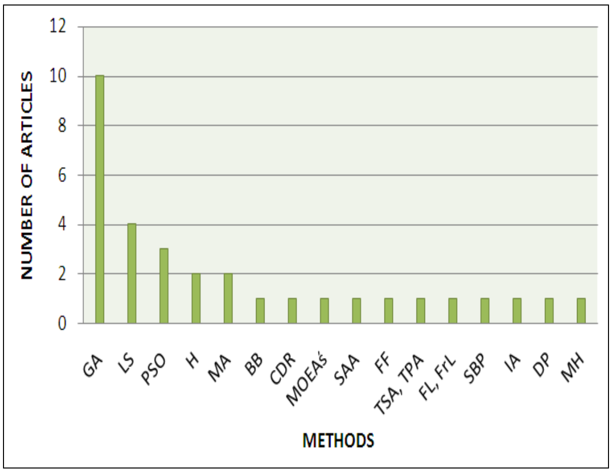

Figure 2 The most commonly used methods of JSSP in order of frequency

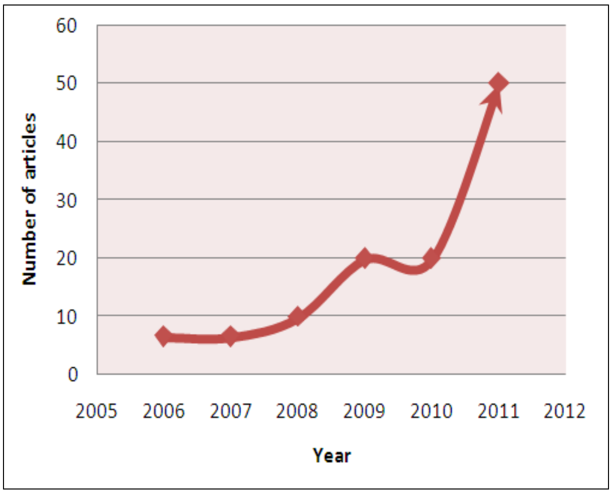

Figure 3 Frequency of research articles dealing with the JSSP over the last 5 years

SUMMARY

Based on the results of the presented study can be stated that Job shop scheduling problem is one of the topical problems in operations research, which is continuously being updated in accordance with the results of newest approaches.

The intention of this paper was to provide an overview of one class of a large group of job shop scheduling problem. In each its part (especially in the previous section), is offered a brief literature of works that dealt with the particular approaches. A part of the main objectives of this study, considerable attention has been paid to the concept of classification of job shop problems and algorithms classification that are pertinent to solve specified problems.

The results from the mapping of important trends in developing new methods can be used as a reference for future research in this area.

References

[1] Brucker P., Kravchenko S.A., and Sotskov Y.N.. Preemptive job-shop scheduling problems with a fixed number of jobs, (1990), Math. Methods Oper. Res., 49(1):41-76.

[2] Carlier, J., and Pinson, E., A practical use of Jacksonś preemptive schedule for solving the job-shop, (1990), Annals of Operations Research 26, 269-287

[3] Carlier, J., and Pinson, E., An algorithm for solving the job shop problem, (1989), Management Science Vol. 35 2, pp. 164-176

[4] Fisher, M.L., Optimal solution of scheduling problems using Lagrange multipliers, Part I, (1973), Operation Res., vol. 21, pp. 1114-1127.

[5] Fox, M.S and Smith S. F, ISIS- A knowledge-based system for factory scheduling, (1984),Expert Syst., vol. 1, pp. 25-49

[6] Graves, S. C., A review of production scheduling. (1981), Operation Research, 29, 646-675

[7] Groover, M. P., Automation Production Systems and Computer-Integrated Manufacturing, 3rd edition, (2007), ISBN-13: 978-0132393218, August 3,

[8] Kusiak A., Intelligent Manufacturing System, (1990), Englewood Cliffs, NJ: Prentice-Hall

[9] Lageweg, B.J., Lenstra J.K. , and Rinnooy Kan A.H.G., Computational complexity of discrete optimization problems, (1979), Ann. Discrete Math., 4:121-140.

[10] Mackay J., Elder P.A., Porteous D.J., Steel C.M., Hawkins R. A., Going J.J., Chetty U., Partial deletion of chromosome 11p in breast cancer correlates with size of primary tumor and estrogen receptor level, (1988), Br J Cancer 58: 710-714

[11] Mastrolilli, M., Gambardella, L.M., Effective neighborhood functions for the flexible job shop problem, (1996),Journal of Scheduling, Vol. 3, pp3-20.

[12] Mati Y., Dauzère-Pérès S., Lahlou CH., 2011, A general approach for optimizing regular criteria in the job-shop scheduling problem, (2011), European Journal of Operational Research 212, 33–4

[13] Mati, Yazid, Dauzère-Pérès, Stèphane, Lahlou, Chams, A general approach for optimizing regular criteria in the job-shop scheduling problem, (2011), European Journal of Operational Research 212, 33–4

[14] Modrak, V. "Case on Manufacturing Cell Formation Using Production Flow Analysis", World Academy of Science, Engineering and Technology 2009, pp. 49-95.

[15] Modrak, V. and Pandian R.S. (2012) Operations Management Research and Cellular Manufacturing Systems: Innovative Methods and Approaches. IGI Global, Hershey USA.

[16] Muth, J. F. and Thompson, G. L., Industrial scheduling, (1963), Prentice-Hall, Englewood Cliffs, NJ

[17] Pezzella, F., Morganti, G., Ciaschetti, G., "A genetic algorithm for the flexible job-shop scheduling problem", (2008), Computers and Operations Research, Vol. 35, No. 10, pp3202-3212.

[18] Schmenner R. W., "Production/Operations Management: Concepts and Situations," 2nd Ed. Science Research Assoc., Chicago (1984).

[19] Dauzère-Pérès, S. and Paulli, J. An Integrated Approach for Modeling and Solving the General Multiprocessor Job-Shop Scheduling Problem Using Tabu Search, (1997), Annals of Operations Research, vol 70,281-306.

[20] Viviers, F. (1983) A Decision Support System for Job-Shop Scheduling, European Journal of Operational Research, Sept, 14(1), 95-103.

[21] Brucker, P., Schile, R.: Job-shop scheduling with multi-purpose machines, Computing. Computing 45(4), 369–375 (1990)