The University of Bradford Institutional Repository

Link to publisher's version: *http://dx.doi.org/10.2202/1934-2659.1602*

Citation: Maqsood S, Khan MK and Wood AS (2011) Novel heuristic for low-batch manufacturing process scheduling optimisation with reference to process engineering. Chemical Product and Process Modeling. 6(2): Article 8.

Chemical Product and Process Modeling

Volume 6, *Issue* 2 2011 *Article* 8

CAPE FORUM 2011

Novel Heuristic for Low-Batch Manufacturing Process Scheduling Optimisation with Reference to Process Engineering

S. Maqsood, *University of Bradford*
M. K. Khan, *University of Bradford*
A. S. Wood, *University of Bradford*

Recommended Citation:
Maqsood, S.; Khan, M. K.; and Wood, A. S. (2011) "Novel Heuristic for Low-Batch Manufacturing Process Scheduling Optimisation with Reference to Process Engineering," *Chemical Product and Process Modeling*: Vol. 6: Iss. 2, Article 8.
DOI: 10.2202/1934-2659.1602
Available at: http://www.bepress.com/cppm/vol6/iss2/8

Novel Heuristic for Low-Batch Manufacturing Process Scheduling Optimisation with Reference to Process Engineering

S. Maqsood, M. K. Khan, and A. S. Wood

Abstract

Scheduling is an important element that has a major impact on the efficiency of all manufacturing processes. It plays an important role in optimising the manufacturing times and costs resulting in energy efficient processes. It has been estimated that more than 75% of manufacturing processes occur in small batches. In such environments, processes must be able to perform a variety of operations on a mix of different batches. Batch-job scheduling optimisation is the response to such low batch manufacturing problems. The optimisation of batch-job process scheduling problem is still a challenge to researchers and is far from being completely solved due to its combinatorial nature. In this paper, a novel hybrid heuristic (HybH) solution approach for batch-job scheduling problem is presented with the objective of optimising the overall Makespan (C_{max}). The proposed HybH is the combination of Index Based Heuristic (IBH) and the Finished Batch-Job (FBJ) process schedule. The heuristic assigns the first operation to a batch-job using IBH and the remaining operations on the basis FBJ process schedule. The FBJ process schedule gives priority to the batch-job with early finished operations, without violating the constraints of process order. The proposed HybH is explained with the help of a detailed example. Several benchmark problems are solved from the literature to check the validity and effectiveness of the proposed heuristic. The presented HybH has achieved batch-job process schedules which have outperformed the traditional heuristics. The results are encouraging and show that the proposed heuristic is a valid methodology for batch process scheduling optimisation.

KEYWORDS: process scheduling, optimisation, batch-job, makespan, hybrid heuristic (HybH), index based heuristic (IBH), small batches, finished batch-job (FBJ)

1. Introduction

The recent trends in process manufacturing show that low unit cost and high quality products no longer solely define an efficient manufacturing system. An efficient manufacturing system represents less waste in its processes. According to Melton (2005), waste is an activity which does not add value to a process. The renowned seven wastes concept of Toyota focuses on improvement of overall process and customer value. The *Mura* aspect of the waste concept includes the scheduling problem and its impact on the efficiency of manufacturing process. Considering the market trends and requirements, scheduling organizes the simultaneous execution of several jobs using flexible resources available in a process, which becomes a complex problem to solve (Noor, 2007). Hence, scheduling is ultimately responsible for an efficient manufacturing process. Its efficiency and failures will therefore highly condition relationship with its customers (Lopez and Roubellat, 2008). Within companies, this scheduling function has always been present, but currently it faces increasingly complex scenarios because of the large number of variety batch-jobs that must be executed simultaneously with shorter manufacturing times (Lopez and Roubellat, 2008). Manufacturing systems operate under constant pressure due to the unpredictability in demand and the ever decreasing product life cycles and are thus finding it hard to cope with these challenges. Process manufacturing sectors are also facing these challenges (Tariq, 2008) and the resolution of the scheduling problem will lead to benefits of reduced wastes in processes, reduced material handling cost, reduced setup times and lower Work-In-Process (WIP) inventory, in all industries including process.

2. The scheduling problem

An efficient scheduling system is an essential part of any manufacturing environment and depends on the scheduling scenario (Noor and Khan, 2007; Janiak and Janiak, 2011). In literature, various researchers (Roy and Sussmann, 1964; Adams, Balas et al., 1988; Jones and Rabelo, 1998; Pinedo, Chao et al., 1998; Jain and Meeran, 1999; Blazewicz, Ecker et al., 2005; Morshed, 2006; Noor, 2007; Zhang and Wu, 2010) have discussed several mathematical models with the objective function of minimizing Makespan ($\boldsymbol{C_{max}}$). This objective is also considered in this paper because it normally performs well on average with respect to criteria such as due date compliance, total completion time, total tardiness, total flow time and maximum lateness. The key criterion has been identified as due date compliance, which is considered to be the most important factor for decision making within industry.

Manufacturing scheduling consists of a set of different resources that perform operations in a process on a batch-job. Each batch-job has a specified processing order through the resources with certain processing times. The batch-job shop environment considered in this paper *does not* allow pre-emption or interruption during a process. In a batch-job shop environment, $\boldsymbol{N}$ is the number of batch-jobs, each with number of operations '$\boldsymbol{O_M}$', which have to be processed on a set of $\boldsymbol{M}$ manufacturing sub-processes. The '$\boldsymbol{O}_{\mathbf{M}}$' is equal to the number of manufacturing sub-processes $\boldsymbol{M}$ $(\boldsymbol{O_M} = \boldsymbol{M})$ with predetermined order or constraint for a given span of time. At any time, only one operation is possible in a single sub-process. This problem is known as NP-hard combinatorial problem (Morshed, 2006; Zhan, Qiu et al., 2009), and can be applied to a wide range of process manufacturing environments. For the last 50 years, researchers have developed and applied various heuristics and techniques to reduce the gap between two operations in a process on a batch–job. The dispatching rules are the most common heuristics used for solving process scheduling problems. Based on predefined criteria, these heuristics select a batch-job to be processed from a queue of jobs. Due to their ease of implementation and substantially reduced computational requirements these approximation-based heuristics are very popular techniques in scheduling (Morshed, 2006). Their importance is derived from the fact that these techniques generate active schedules or, in other words, schedules in a search space where an optimum value can be achieved (Noor, 2007). In the cited literature, the performance of well known heuristics depends on the size of a problem which means one single heuristic cannot solve any type and size of a problem. These factors have encouraged the researchers to develop efficient heuristics which not only outperform traditional heuristics, but also provide a heuristic which can be applied to different problems. In this paper, a novel HybH is presented and is based on the combination of Index Based Heuristic (IBH) and Finished Batch-Job (FJB) process schedule. The IBH itself is a novel way of evolving a process schedule. The HybH is tested for benchmark problems and compared with traditional heuristics.

3. Description of the novel hybrid heuristic

The set of steps followed in the development of proposed heuristic is summarized in the flowchart shown in Figure 1. The proposed HybH consists of the following five steps:

> *Step 1:* For given processing plan and processing times initially, the HybH converts the processing times into Index Values (IVal) of each batch-job. Once the conversion is done, the batch-jobs for each operation are then sorted on the basis of ascending IVal for each operation. The sorted batch-jobs are then processed for the first operation.

Step 2: For the first operation, the HybH tries different combinations (equals the number of operations) for the best possible schedule and therefore evolves a schedule for each combination. During a schedule evaluation, output date such as batch-job processed, next process due, start time for each process, finish time for each process are recorded. When an operation is completed, the algorithm deletes that operation from the list of all possible operations for a batch-job.

Step 3: For the remaining processes, the HybH evolves schedules using the proposed FBJ process schedule. The FBJ process schedule assigns the next operation on the batch-job with the early finished time already recorded without violating the precedence constraints.

Step 4: Repeat the procedure until all batch-jobs are processed for all operations on the basis of earlier finished time.

Step 5: Find the maximum of the maximum of finish times on all processes (Makespan).

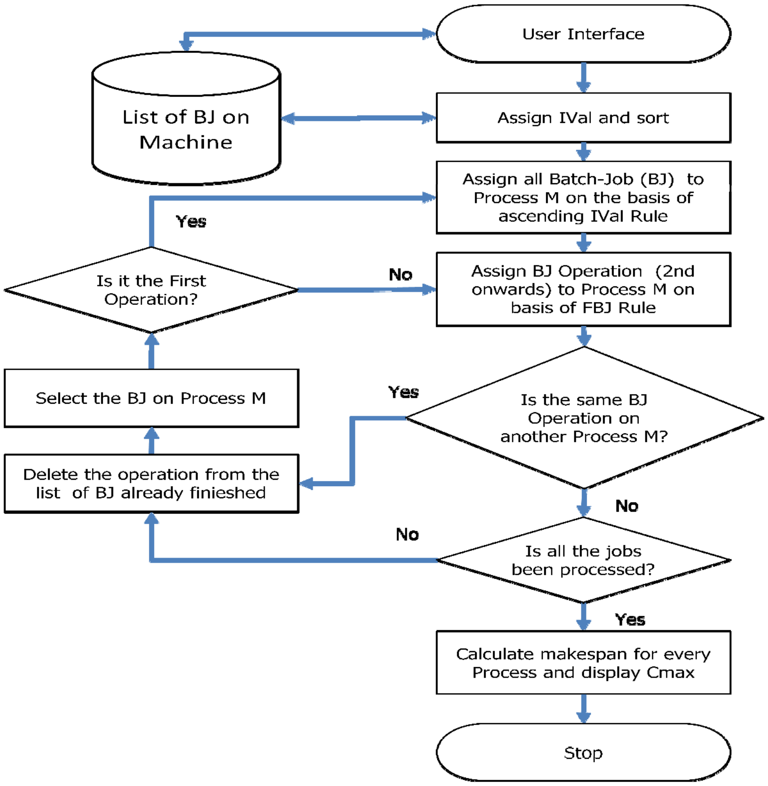

Figure 1: Proposed heuristic for batch-job scheduling problem

A simple example of four batch-jobs and four processes (Noor, 2007) is taken from literature and is used illustrate HybH. Table 1 shows the process plan.

For example, J_2 has three operations O_1, O_2, O_3 and O_4 on M_2 (with a processing time unit of 1), M_4 (with a processing time unit of 5), M_1 (with a processing time unit of 3) and M_3 (with a processing time unit of 2), respectively.

Process Plan								
	O_1		***O_2***		***O_3***		***O_4***	
Jobs	**M**	**PT**	**M**	**PT**	**M**	**PT**	**M**	**PT**
J_1	2	3	1	3	4	2	3	2
J_2	**2**	**1**	**4**	**5**	**1**	**3**	**3**	**2**
J_3	4	3	1	2	3	3	2	4
J_4	4	2	2	4	1	3	3	3

Table 1: Process Plan [Noor (2007)]

At first step, all the operations which are available to be scheduled are assigned with an Index Value (IVal). Table 2 shows the index based representation of the problem. The IVal for any batch-job is calculated by adding all the processing times for a job and then dividing it by the processing time of remaining operations. For example in J_2, the index value for operation O_1 is 0.09091 [1/(1+5+3+2)], for operation O_2 it is 0.5 [5/(5+3+2)], for operation O_3 it is 0.6 [3/(3+2)] and for operation O_4 it is 1 [2/2], respectively.

Process Plan: Index Value Based								
	O_1		***O_2***		***O_3***		***O_4***	
Jobs	**M**	**IVal**	**M**	**IVal**	**M**	**IVal**	**M**	**IVal**
J_1	2	0.0909	1	0.4286	4	0.5	3	1
J_2	**2**	**0.09091**	**4**	**0.5**	**1**	**0.6**	**3**	**1**
J_3	4	0.25	1	0.2222	3	0.4286	2	1
J_4	4	0.16667	2	0.4	1	0.5	3	1

Table 2: Index Based Representation of Process Plan

The hybrid heuristic take the process plans (see Table 2) and sorts the batch-jobs for each operation on the basis of ascending index values as shown in

Table 3. Assuming that at start all processes are available at time zero units, the operation with ascending value is then selected followed by the next ascending value and so on. These operations are then sequenced to be processed.

Sorted Process Plan: In Index Value											
O_1			***O_2***			***O_3***			***O_4***		
Jobs	**M**	**IVal**	***Jobs***	**M**	**IVal**	***Jobs***	**M**	**IVal**	***Jobs***	**M**	**IVal**
J_2	2	0.0909	***J_3***	1	0.2222	***J_3***	3	0.4286	***J_1***	3	1.0000
J_4	4	0.1667	***J_4***	2	0.4000	***J_1***	4	0.5000	***J_2***	3	1.0000
J_3	4	0.2500	***J_1***	1	0.4286	***J_4***	1	0.5000	***J_3***	2	1.0000
J_1	2	0.3000	***J_2***	4	0.5000	***J_2***	1	0.6000	***J_4***	3	1.0000

Table 3: Sorted Process Plan on the basis of ascending IVal

The sorted batch-jobs (shown in Table 3) for first sub-process or Operation O_1 are batch-jobs [J_2, J_4, J_3, J_1] which are to be processed. For example, J_2 was the first finished operation on process M_2; hence J_2 Operation O_2 is scheduled first followed by J_4 (O_2), J_1 (O_2) and J_3 (O_2). The rest of the schedule (active schedule) is evolved through the same procedure. Figure 2 shows the Gantt chart of the final schedule obtained from HybH.

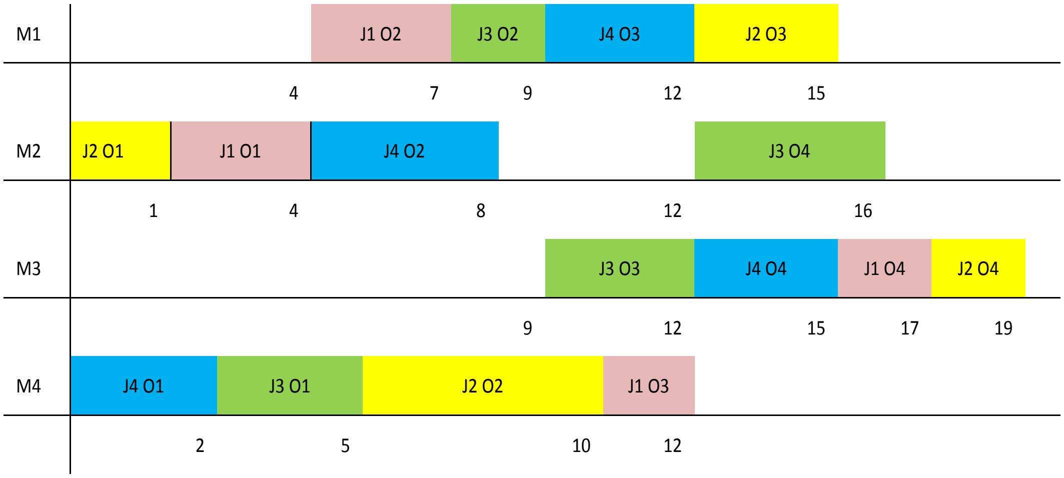

Figure 2: Gantt chart showing the evolved schedule from HybH

4. Implementation

The HybH has been implemented in MATLAB 7 and on an Intel(R) Core 2 Duo processor (2.00GHz). The data required for the algorithm was in the form of processing times and process plan in spread sheets. These spread sheets were used for inputting data to MATLAB. The traditional heuristics (used for comparison with the HybH) were simulated in LAKIN scheduling software.

5. Benchmark problems and traditional heuristics

To gauge their strength and comparative merits, all new heuristics are tested against published benchmark problems. These benchmark problems provide a common standard on which scheduling algorithms can be tested and compared. As the benchmark problems are of different dimensions and grades of difficulty, it is simple to determine the capabilities and limitations of a given method by testing it on these instances. These benchmark problems have been developed by various researchers (Fisher and Thompson (1963) - FT; Carlier (1978) - CAR; Lawrence (1984) - LA; Adams et al., (1988) - ABZ; Applegate and Cook (1991) - ORB; Storer et al., (1992)- SWV; Yamada and Nakano (1992) – YN and Taillard (1993). In this paper, FT (06, 10) and LA (01, 06, 11, 12, 26 and 36) benchmark problems are used as test beds to gauge effectiveness of the proposed HybH over traditional heuristics.

Table 4 and Table 5 present the computational results of the proposed HybH. These two tables also provide comparative analysis of the HybH with well known traditional heuristics from literature: Shortest Processing Time (SPT), Longest Processing Time (LPT), First Come First Serve (FCFS), Earliest Due Date (EDD), Critical Ration (CR), Minimum Slack (MS), and Weighted Shortest Processing Time (WSPT). The comparisons are made using Relative Deviation (RD) measure or Mean Relative Error (MRE) or Percent GAP (% GAP). The measure % GAP is the deviation of the Makespan value obtained by a particular heuristic from the optimum or global Makespan. It represents a measure of quality of the best global Makespan. The % GAP is calculated from the best known global Lower Bound (LB) or optimum Makespan, and the Upper Bound (UB) Makespan achieved by a particular heuristic being analysed using the following relative deviation formula

$$\%\,GAP = \left\lfloor \frac{UB - LB}{LB} \right\rfloor \times 100$$

Morshed (2006) reported that in the analyses based on % GAP, the traditional heuristics have achieved results extremely quickly, but they are of very poor quality (GAP from optimum schedule can be as great as 74%) and in general, the solution quality degrades as the problem dimensionality increases.

6. Computational experiments and results

Using the proposed HybH and the traditional heuristics the Makespan values have been obtained for the defined benchmark problem sets of Fisher and Thompson, (1963) – FT and Lawrence, (1984) – LA, shown in Table 4. For example, the Makespan value obtained by FCFS heuristic for FT06 (6x6 – six batch-jobs and six operations) case is 65 with 18.2% GAP or relative deviation from optimum. Although, the traditional heuristics are computationally fast, none of them have achieved the optimum or near optimum Makespan. Thus the % GAP for FCFS rule is 18.2% from the optimal LB value and clearly indicates that FCFS rule for FT06 is inefficient. Looking at the FT06 results, it can be seen that average Makespan for the seven heuristic rules is 70 with a GAP of 27%. The best result recorded a Makespan of 63 and a GAP of 14.5% (for EDD rule), whilst the worst result is a Makespan of 81 and a GAP of 47.3% (for CR rule). For the FT10 (10x10) benchmark problem the heuristic rule's performance was different. The best Makespan achieved for FT10 was 1168 with 25.6% GAP (by LPT rule) and the worst result achieved was 1338 with 43.87% GAP (by the SPT and WSPT rules). The Proposed HybH in comparison has performed much better against the test bed except for problem FT10.

Test Bed	**Fisher and Thompson (1963) - FT**				**Overall Mean GAP%**
Problem	**FT06 (*Opt=55)**		**FT10 (Opt=930)**		
Instances	**6x6**	**GAP%**	**10x10**	**GAP%**	
FCFS	65	18.2%	1184	27.3%	22.7%
LPT	67	21.8%	**1168**	25.6%	23.7%
SPT	73	32.7%	1338	43.9%	38.3%
CR	81	47.3%	1181	27.0%	37.1%
EDD	63	14.5%	1246	34.0%	24.3%
MS	67	21.8%	1168	25.6%	23.7%
WSPT	73	32.7%	1338	43.9%	38.3%
Average	***70***	*27.0%*	***1232***	*32.5%*	*29.7%*
Minimum	***63***	*14.5%*	***1168***	*25.6%*	*20.1%*
Maximum	***81***	*47.3%*	***1338***	*43.9%*	*45.6%*
HybH	**61**	10.9%	1175	26.3%	18.6%

*The Optimum Makespan value

Table 4: HybH vs. Traditional Heuristics for FT Benchmark Problems (Fisher and Thompson, 1963)

For FT06 problem it achieved a Makespan of 61 (with 10.9% GAP) versus the best perform one EDD (63 with 14.5% GAP). For the FT10 (10x10) results, it did not make the best performance. Here, the best value was achieved by LPT and WSPT (1168 with 25.6% GAP) versus HybH (1175 with 26.3% GAP). This is due to the fact that the LPT and WSPT heuristics suits FT10, because of two main reasons. Firstly, Fisher and Thompson, (1963) have assigned lower number of machines to earlier operations and higher number of machines for later operations. Secondly, the first operation comparatively has larger processing times and HybH sorts the first operation on the basis of ascending index values. Hence the FT10 has been reported in literature as "notoriously" hard because it is different from other benchmark cases. It might be fruitful if the proposed heuristic solves this FT10 problem with first operation job order sorted on the basis of larger index values. However, the HybH was close to the minimum value and certainly performed better then the results of the traditional heuristics, as shown in Table 4.

Test Bed	**Lawrence (1984) - LA**												**Overall Mean GAP%**
Problem	**La01 (Opt=666)**		**La06 (Opt=926)**		**La11 (Opt=1222)**		**La12 (Opt=1039)**		**La26 (Opt=1218)**		**La36 ((Opt=1268)**		
Instances	**10x5**	**GAP %**	**15x5**	**GAP %**	**20x5**	**GAP %**	**20x5**	**GAP %**	**20x10**	**GAP %**	**15x15**	**GAP %**	
FCFS	772	15.9%	**926**	0.0%	1272	4.1%	**1039**	0.0%	1505	23.6%	1516	19.6%	10.5%
LPT	752	12.9%	**926**	0.0%	1300	6.4%	1167	12.3%	1394	14.4%	1480	16.7%	10.5%
SPT	1122	68.5%	1475	59.3%	1802	47.5%	1439	38.5%	1993	63.6%	2250	77.4%	59.1%
CR	979	47.0%	1140	23.1%	1792	46.6%	1401	34.8%	2069	69.9%	2229	75.8%	49.5%
EDD	865	29.9%	1024	10.6%	1272	4.1%	**1039**	0.0%	1430	17.4%	1550	22.2%	14.0%
MS	752	12.9%	**926**	0.0%	1300	6.4%	1167	12.3%	1394	14.4%	1480	16.7%	10.5%
WSPT	1122	68.5%	1475	59.3%	1802	47.5%	1439	38.5%	1993	63.6%	2250	77.4%	59.1%
Average	***909***	*36.5%*	***1127***	*21.7%*	***1506***	*23.2%*	***1242***	*19.5%*	***1683***	*38.1%*	***1822***	*43.7%*	*30.5%*
Minimum	***752***	*12.9%*	***926***	*0.0%*	***1272***	*4.1%*	***1039***	*0.0%*	***1394***	*14.4%*	***1480***	*16.7%*	*8.0%*
Maximum	***1122***	*68.5%*	***1475***	*59.3%*	***1802***	*47.5%*	***1439***	*38.5%*	***2069***	*69.9%*	***2229***	*75.8%*	*59.9%*
HybH	**700**	**5.1%**	**926**	**0.0%**	**1272**	**4.1%**	**1039**	**0.0%**	**1358**	**11.5%**	**1453**	**14.6 %**	**5.9%**

Table 5: HybH vs. Traditional Heuristics for LA Benchmark Problems (Lawrence, 1984)

To further explore the strength and weaknesses of the proposed heuristic, the HybH has been tested against benchmark cases developed by Lawrence (1984). These cases are of various instances, which range from 10x5, 15x5, 20x5, 15x10 to 15x15, as shown in Table 5. Referring to the results in Table 5, for the traditional heuristics the FCFS has achieved optimum for two cases: LA06 and LA12. Whilst, the LPT and MS have achieved the optimum for LA06 and the EDD has achieved the optimum for LA12. In comparison, the proposed HybH has not only achieved the optimum values for LA06 and LA12, it also achieved the new LB values for all test bed cases.

Table 5 also shows the overall mean % GAP taken across the LA-problems. The proposed HybH has lesser % GAP value of 6% in comparison with the best traditional heuristics, which have a overall mean GAP value of 10.5%, (LPT and MS rules). Hence, the HybH has reduced the overall % GAP by 77.9%, which reflects a considerable gain in process efficiency.

In summary, the proposed HybH has performed consistently well across the test bed FT and LA benchmark problems and can be applied to any size of problem. However, in the case of traditional heuristics, the performance of each heuristic depended on the type and size of the benchmark problem.

7. Conclusion

The majority of the processing time based heuristics with the objective function of optimising the Makespan reported in literature are computationally very fast but their relative difference (GAP) from the optimum LB is as large as 75%. Furthermore, for traditional heuristics, no single rule performed well across all the test bed problems. The proposed HybH overcame the deficiencies in the traditional heuristics for manufacturing process scheduling. The novel HybH has performed well across all the test bed benchmark problems and successfully achieved new LB, optimal or near optimal solutions for batch-job process scheduling problems. It has reduced the % GAP in each test bed problem and the overall mean % GAP by a considerable amount. Future work will focus on real scheduling problems, including process manufacturing.

8. References

ADAMS, J., BALAS, E. and ZAWACK, D. (1988). The shifting bottleneck procedure for job shop scheduling. *Management Science* 34(3): 391-401.

APPLEGATE, D. and COOK, W. (1991). A computational study of the job-shop scheduling problem. *Informs Journal of Computing* 3(2): 149-156.

BLAZEWICZ, J., ECKER, K. and TRYSTRAM, D. (2005). Recent advances in scheduling in computer and manufacturing systems. *European Journal of Operational Research* 164(3): 573-574.

FISHER, H. and THOMPSON, G. L. (1963). Probabilistic learning combinations of local job shop scheduling rules. Industrial Scheduling. J. F. Muth and G. L. Thompson Eds. *Englewood Cliffs, New Jersey.*, Prentice Hall. 15.

JAIN, A. S. and MEERAN, S. (1999). Deterministic job-shop scheduling: Past, present and future. *European Journal of Operational Research* 113(2): 390-434.

JANIAK, A. and JANIAK, W. (2011). Single-processor scheduling problem with dynamic models of task release dates. *IEEE Transactions on Systems Man and Cybernetics Part a-Systems and Humans* 41(2): 264-271.

JONES, A. and RABELO, J. C. (1998) Survey of job shop scheduling techniques. NISTIR, National Institute of Standards and Technology, Gaithersburg, MD.

LAWRENCE, S. (1984). Resource Constrained Project Scheduling: An Experimental Investigation of Heuristic Scheduling Techniques (Supplement), . *Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburg, 1984*.

LOPEZ, P. and ROUBELLAT, F., Eds. (2008). Production Scheduling. *London*, John Wiley.

MELTON, T. (2005). The benefits of lean manufacturing - What lean thinking has to offer the process industries. *Chemical Engineering Research & Design* 83(A6): 662-673.

MORSHED, M. S.(2006). A hybrid model for job shop scheduling. PhD, University of Birmingham, UK

NOOR, S.(2007). Operational scheduling of traditional and flexible manufacturing system using genetic algorithms, artificial neural networks and simulation. PhD, University of Bradford

NOOR, S. and KHAN, M. K. (2007). Hybrid scheduling system for flow shop and job shop scheduling problem. *23rd International Conference of CAD/CAM, Robotics & Factories of the Future Conference,* Bogota, Colombia.

PINEDO, M., CHAO, X. and PINEDO, M. L., Eds. (1998). Operations scheduling with applications in manufacturing and services. *Singapore*, McGraw-Hill.

ROY, B. and SUSSMANN, B. (1964). Les Problems d'Ordonnancement avec Contraintes Disjonctives. *Note D. S. no. 9 bis, SEMA, Paris, France*.

STORER, R. H., WU, S. D. and VACCARI, R. (1992). New search spaces for sequencing problems with application to job shop scheduling. *Management Science* 38(10): 1495-1509.

TAILLARD, E. (1993). Benchmarks for basic job shop scheduling problems. *European Journal of Operational Research* 64(2): 278-285.

TARIQ, A.(2008). Operational design of a Cellular Manufacturing System (CMS). PhD, NUST, Pakistan

YAMADA, T. and NAKANO, R. (1992). A genetic algorithm application to large scale job shop problems. *Parallel Problem Solving from Nature, 2*: 281-290.

ZHAN, Y., QIU, C. H. and XUE, K. (2009). A hybrid genetic algorithm for hybrid flow shop scheduling with load balancing. *Manufacturing Automation Technology* 392-394: 250-255.

ZHANG, R. and WU, C. (2010). A hybrid approach to large-scale job shop scheduling. *Applied Intelligence* 32(1): 47-59.