International Journal of Scientific & Engineering Research, Volume 4, Issue 6, June-2013 1809
ISSN 2229-5518

Optimization Of Flexible Jobshop Scheduling Problem Using Attribute Oriented Mining

Neelesh Kumar Sharma

Abstract— Flexible job-shop scheduling problem (FJSSP) is an extension of the classical job-shop scheduling problem that allows an operation to be processed by any machine from a given set along different routes. It is very important in both fields of production management and combinatorial optimisation. This paper presents a new approach based on attribute oriented mining technique to solve the multi-objective flexible job-shop scheduling problem. Three minimisation objectives - the maximum completion time, the total workload of machines and the workload of the critical machines are considered simultaneously. In this study, a hybrid method is used to assign operations and to determine the processing order of jobs on machines. The objectives are optimised by attribute oriented mining technique which extracts the knowledge from the solution sets to find the near optimal solution of combinatorial optimisation problems. The computational results have shown that the proposed method is a feasible and effective approach for the multi-objective flexible job-shop scheduling problems.

Index Terms— Flexible job shop, attribute oriented mining, optimization problems, multi objective job shop, artificial intelligence,

1 INTRODUCTION

1.1 Scheduling

Scheduling is a important issue in the planning and manufacturing operations. The proper scheduling of machines in an industry can reduces the production hours that contributes to produce goods much faster. In other way scheduling is a decision making process to determine when a job is to be started in a machine and when it is to be completed. A job may have to be processed at various different machines or process centers. Similarly a machine or process centre may have to take different jobs and complete them. The order in which jobs are arranged on a machine or process centre is called sequencing. Scheduling refers to a set of policies and mechanisms to control the order of work to be performed by a system.

Scheduling is the decision-making processes that are used on a regular basis in many manufacturing and service industries. These forms of decision-making play an important role in procurement and production, in transportation and distribution, and in information processing and communication. Scheduling activity in an organization depends upon the availability resources and the methods to allocate limited resources to the activities that have to be done. This allocation of resources has to be done in such a way that the organization optimizes its objectives and achieves its goals. Each activity may have a priority level, an earliest possible starting time and a due date. Objectives can take many different forms, such as minimizing the time to complete all activities, minimizing the number of activities that are completed after the committed due dates, and so on. Based on the requirement of the problem, the objective functions are formulated and weights are given to them according to the priority assigned to each objective function.

1.2 Role Of Planning And Scheduling

Production control department starts its functioning from gathering the work order from the sales department, this order is the starting point for all the activities of production control department concerned with the manufacturing of products. The loading of various work centers is carried out. A copy of the master production schedule is passed to the material control section, the role of material control is to assess the need of material, and take appropriate steps to meet the required demand. The manufacturing process starts after collecting the relevant documents of the each section and verifies the availability of each of the factors at production and starts the production activity .The progress section section will monitor the performance and verifies that requirements of the master production schedule are fulfilled. Any deviation from the schedule are brought to the notice of the concerned persons and corrective actions are taken to keep the deviation at minimum

1.3 Types Of Scheduling Problems

1 .SINGLE MACHINE SCHEDULING

The simplest pure sequencing problem is one which there is a single resource, or machine. As simple as it is, however the single machine case still very important for several reasons .it provides a context in which to investigate many different performance measures and several solution techniques. It is therefore a building block in the development of the comprehensive understanding of the scheduling concepts, an understanding that ultimately facilitate the modeling of complicated systems.

2 .FLOW SHOP SCHEDULING

There is more than one machine and each job must be processed on each of the machines – the number of operations for each job is equal with the number of machines, the *jth* operation of each job being processed on machine *j*; In each job exactly one operation for every machine, all jobs go through all the machines in the same order.

Any group of machines served by a unidirectional, non cyclic conveyor wound is considered as a flow shop. Flow shop is one in which all the jobs follow essentially the same path from one machine to another. In flow shop machines are arranged in series and job begin processing on an initial machine, proceed through several intermediary machines, and completed in the final machine. Each machine will take up the jobs in a sequence to perform the operation required. The sequence of jobs for all the machines is same.

3. JOB SHOP SCHEDULING

In this job can be processed on machines in any order. The general job shop is one, in which *n* jobs are to be processed by m machines. Each job will have set of constraints on the order in which machines can processed and a given processing times on each machines. Jobs may not require all *m* machines and they may have to visit some machines more than once. In a job shop, specific machine order restriction is not imposed in each job.

1.4 Introduction To Flexible Job Shop Scheduling Problem

Flexible job shop problem(FJSP) is an extension of the classical job shop problem(JSP) its objective is to minimizing the makespan, the operations to be processed is fixed by finding the optimal sequence, were as in flexible job shop problem it allows flexibility of processing an operation on the available machines, because of that it becomes complex to determine the assignment of operations to the machines, its objective is to minimizing the makespan by balancing the workloads on the machines by finding operation sequences with machine selection.

Flexible job shop breaks the restriction of unique allocation of each operation to be processed by several different machines, thus making the job-shop scheduling problem close to the actual production system. The processing time of each operation on the available machines is fixed and known corresponding to the operations for each job the set-up times between operations are either negligible or included in processing times and each machine is continuously available from time zero there are no precedence constraints among operations of different jobs, each operation cannot be interrupted, each machine can process at most one operation at a time.

For solving the FJSP two types of approaches have been used, hierarchical approaches and integrated approaches. In hierarchical approaches assignment of operations to machines and the sequencing of operations on the machines are treated separately, i.e. assignment and sequencing are considered independently, where as in integrated approaches, assignment and sequencing are not differentiated. Hierarchical approaches are based on the idea of decomposing the original problem in order to reduce its complexity by solving the routing and the scheduling as two sub-problems.Integrated approaches considers both assignment and sequencing at the same time.These integrated approaches pave the way for formulating the multi-objective functions. In single objective functions makespan was only considered, where as in multi-objective function tardiness, due-date, critical machine work load and total work load are considered. Multi-objective functions are formulated based on the requirement of the problem. Flexible job shop problem can be differentiating in two kinds; i.e. total flexibility problem and partial flexibility problem.

1. Total flexibility: in this case all operations are processed on all the machines available.

2. Partial flexibility: in this case some operations are only processed on all the available machines, and some operations are restricted to processed on available machines.

1.5 Introduction To Particle Swarm Algorithm

Particle swarm optimization (PSO) is an evolutionary computation technique proposed by Kennedy and Eberhart in 1995. The particle swarm concept was motivated from the simulation of social behavior. The original intent was to graphically simulate the graceful but unpredictable choreography of a bird flock. The PSO algorithm mimics the behavior of flying birds and their means of information exchange to solve optimization problems. PSO has been introduced as an optimization technique in real-number spaces. But many optimization problems are set in a discrete space. Typical examples include problems that require ordering and route planning, such as in scheduling and routing problems. In this paper we introduce a method of converting domain to discrete domain for PSO and use it to assign operations to machines.

3.1. Standard PSO algorithm

PSO is an evolutionary algorithm which the system is initialized with a population (named swarm in PSO) of random solutions. Each individual or potential solution, named particle, flies in the D-dimensional problem space with a velocity which is dynamically adjusted according to the flying experiences of its own and its colleagues. During the past years, researchers have explored several models of PSO algorithm. In this paper, we use the global model equations as follows (Shi & Eberhart, 1999):

$$V_{id}=W*V_{id}+C1*Rand()*(p_{id}-X_{id})+C2*rand()*(P_{gd}-X_{id}), \qquad (1a)$$

and

$$X_{id}=X_{id}+V_{id}, \qquad (1b)$$

where V_{id}, called the velocity for particle i, represents the distance to be travelled by this particle from its current position, X_{id} represents the particle position, P_{id} which is also called pbest (local best solution), represents ith particle's best previous position, and P_{gd}, which is also called gbest (global best solution), represents the best position among all particles in the swarm. W is inertial weight. It regulates the trade-off between the global exploration and local exploitation abilities of the swarm. The acceleration constants C1 and C2 represent the weight of the stochastic acceleration terms that pull each particle toward pbest and gbest positions. Rand() and rand() are two random functions in the range [0,1].

For Eq. (1a), the first part represents the inertia of previous velocity. The second part is the ‘cognition’ part, which represents the private thinking by itself. The third part is the ‘social’ part, which represents the cooperation among the particles (Kennedy, 1997). The process of implementing the PSO algorithm is as follows:

ISSN 2229-5518

(1) Initialize a swarm of particles with random positions and velocities in the D-dimensional problem space.

(2) For each particle, evaluate the desired optimization fitness function.

(3) Compare particle's fitness value with particle's pbest. If current value is better than pbest, then set pbest value equal to the current value, and the pbest position equal to the current position in D-dimensional space.

(4) Compare fitness evaluation value with the best swarm's fitness obtained so far. If current value is better than gbest, then reset gbest to the current particle's fitness value.

(5) Change the velocity and position of the particle according to Eqs. (1a) and (1b) respectively.

(6) Loop to step (2) until a termination criterion is met, usually a sufficiently good fitness or a specified number of generations.

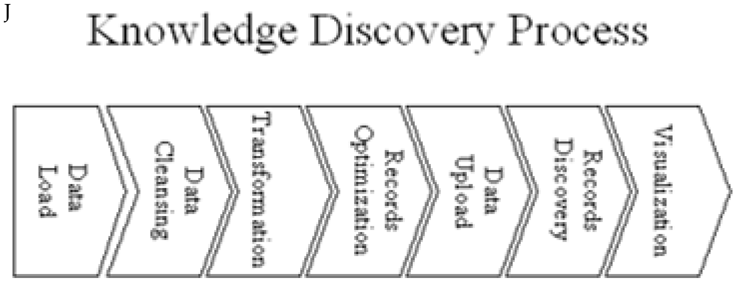
In PSO, each particle of the swarm shares mutual information globally and benefits from the discoveries and previous experiences of all other colleagues during the search process. PSO requires only primitive and simple mathematical operators, and is computationally inexpensive in terms of both memory requirements and time.

1.5 Introduction To Data Mining

A new generation of techniques and tools are required to assist humans in intelligently analyzing voluminous data for pieces of useful knowledge. Knowledge Discovery in Databases (KDD) and Data Mining integrate DBMS and artificial intelligence technologies to assist humans in analyzing large quantities of data. Knowledge Discovery in Databases is defined as “the non-trivial process of identifying valid, novel, potentially useful, ultimately understandable patterns in data”. In other words, KDD involves the evaluation and interpretation of patterns to determine what constitutes knowledge. Data mining is an application, under human control, of low-level induction algorithms that are used to extract patterns from data in specific categories. The difference between KDD and data mining is that data mining focuses on the implementation of induction algorithms and KDD focuses on automated knowledge discovery processes.

Most data mining algorithms are derived from machine learning, pattern recognition and statistics. These algorithms include classification, clustering and graphical models. The primary goals of knowledge discovery are prediction and description. Prediction involves using variables or fields in the database to predict unknown or future values of other variables or attributes. For example, some of its characteristics, such as size, style, location and number of rooms, can predict the monetary value of a house. Description focuses on finding human-interpretable patterns describing the data, such as finding patterns for “good” schedules.

Fig1.



2 LITERATURE REVIEW

2.1 Literature Review On Fjssp

IJSER Bruker and Schlie (1990) were the first to address the FJSSP problem, they develop a polynomial algorithm for solving the flexible job-shop problem with two jobs. For solving the realistic case with more than two jobs, two types of approaches they have proposed. Hierarchical approaches and integrated approaches. In hierarchical approaches assignment of operations to machines and the sequencing of operations on the resources or machines are treated separately, i.e. assignment and sequencing are considered independently, where as in integrated approaches, assignment and sequencing are not differentiated.[Weijun Xia, Zhiming Wu, 2005]

Brandimarte (Brandimarte, 1993) was the first to use this decomposition for the FJJSP. He solved the assignment problem using some existing dispatching rules and then focused on the resulting job shop subproblems, which are solved using a tabu search heuristic.[Haipeng Zhang, and Mitsuo Gen,2005].

Haipeng Zhang, and Mitsuo Gen has proposed a new multistage operation-based representation of GA (moGA) approach is proposed to solve FJJSP considering makespan,critical work machine load and total work load as the objective functions.

Xia and Wu (2005) proposed a hybrid algorithm using particle swarm optimization (PSO) assignment and simulated annealing (SA) scheduling to optimize multi-objective FJSSP, respectively. To our knowledge, research on multi-objective FJSSP is rather limited, and most traditional optimal approaches used only one optimization algorithm for solving multi-objective FJSSP.

Kacem, Hammadi, and Borne (2002a, 2002b) proposed a genetic algorithm controlled by the assigned model which was generated by the approach of localization (AL) to mono-objective and multi-objective FJSSP. They used the integrated approach considering assignment and scheduling at the same time.

Haoxun Chen, Jurgen Ihlow and Carsten Lehmann has proposed an Genetic Algorithm to solve the large class of scheduling problems, for classical job shop and flexible job shop problem considering makespan as the criteria.

Guohui Zhang, Xinyu Shao (2008) the search mechanism of the particle swarm optimization and tabu search is taken full advantage of. An effective solution approach is proposed for solving multi-objective FJSSP. The proposed approach uses PSO to assign operations on machines and to schedule operations on each machine, and TS is applied to local search for the scheduling sub-problem originating from each obtained solution. The objectives which are considered in this paper are to minimize maximal completion time, the workload of the critical machine and the total workload of machines simultaneously.

Nozha Zribi and Pierre Borne (2005) proposed a hybrid genetic algorithm for the flexible job-shop problem under maintenance constraints. They study the flexible job-shop scheduling problem where the machine maintenance has to be performed within certain intervals and hence the machine becomes unavailable during the maintenance periods.

B. Naderi & M. Zandieh & S. M. T. Fatemi Ghomi (2008) were studied flexible job shop scheduling problems with sequence-dependent setup times and preventive maintenance to minimize makespan. Two techniques were presented to integrate production scheduling and Production Management operations. To solve the problem, they proposed four meta-heuristics based on genetic algorithm and simulated annealing.

Imed Kacem, Slim Hammadi, Pierre Borne (2007) proposed an aggregative approach for solving multi-objective optimization This approach makes it possible to construct a set of satisfactory solutions according to the preferences of the decision-maker problems based on the hybridization of fuzzy logic (FL) and evolutionary algorithms (EAs).

Klaus Jansen, Monaldo Mastrolilli, and Roberto Solis-Oba (2000) developed Approximation Algorithms for Flexible Job Shop Problems. The Flexible Job Shop Problem is a generalization of the classical job shop scheduling problem in which for every operation there is a group of machines that can process it. The problem is to assign operations to machines and to order the operations on the machines, so that the operations can be processed with minimum makespan.

Jie Gao a, Mitsuo Gen , Linyan Sun , Xiaohui Zhao (2007) developed a new approach hybridizing genetic algorithm with bottleneck shifting to fully exploit the ''global search ability'' of genetic algorithm and ''the local search ability'' of bottleneck shifting for solving multi objective flexible job shop scheduling problems.

Noureddine Liouance,Ihsen Saad,SlimHammadi and Pierre Borne (2007) has proposed a new approach by the combination ant systems and tabu search optimization for flexible job shop scheduling problem.

F.Pezzella,G.Morganti and G.Ciaschetti has proposed an Genetic Algorithm for framework by the integration of different strategies for the selection of the chromosomes at different stages reproduction, crossover and mutation.

3 Problem Formulation

The flexible job shop scheduling problem, considering the objective functions as minimizing the makespan, and balancing the workload of all the machines and critical workload machine among all the machines available.

3.1 Assumptions For The Fjssp

1 All the machines are available at $t = 0$
2 All the jobs can be started at $t = 0$
3 Each job consists of one fixed sequence of operations.
4 Jobs are independent from each other.
5 Pre-emption is not allowed, once an operation is started on a machine it must be completed before another operation to started on that machine.
6 Each machine can process at most one operation at a time.
7 There are no precedence constraints among the operations of different jobs.
8 Setup time of the machines and the time to move jobs between machines are considered as negligible.
9 Machines may be ideal.

3.2 Problem Formulation

The FJSP can be an extension of the classical JSP; therefore, we can formulate the FJSP based on JSP. The FJSSP can be described as following, Consider a set of n jobs, noted, $J = \{ J_1, J_2, \ldots J_n\}$, every job in the set J has a given number operations, and should be operated on a given machine from a machine set named $M = \{M_1, M_2 \ldots M_m\}$. So, there are n jobs and m machines. In the classical JSP problem, with n jobs and m machines, there are $n \times m$ operations. However, in FJSP problems, there is a problem constraint for the operating process, one operation of a job must be processed by a set of machines in M' □M. Every operation can be processed on different machines, and the process time of each operation is different according to the performance of different machine.

- The detailed definition of the FJSP as follows:

A set of J independent jobs.

Each job J_i can be operated on a given set of machines M_i.

The O_{ij}, represents the j^{th} operation of J_i. The machines set waiting for processing the $O_{i,j}$ noted by M_k □□M.

We use $k_{p,i,j}$, to represent the processing time of $O_{i,j}$ operated on the k^{th} machine.

- The following criteria are to be minimized:

The maximal completion time of machines, i.e., the make span.

The critical machine workload, i.e., the maximum working time spent on any machine.

The total workload of all the machines, which represents the total working time over all machines.

- The FJSS problem is characterized by the following data and notations:

A set of jobs J_i ($i = 1 \ldots n$);

A set of machines m_j ($j = 1 \ldots m$);

The j th operation ($j = 1 \ldots n_i$) of job j_i is denoted by O_{ij};

For each operation O_{ij} the set M_{ij} of machines to perform it is given;

For each machine $m_k \in M_{ij}$ able to execute operation O_{ij}, a processing time p_{ijk} is given.

t_{ijk} : start time of of operation O_{ij} on machine k;

C_{ij} : completion time of the operation O_{ij};

i, h : index of jobs, i, h = 1, 2,. . . ,n;

k: index of machines, k = 1, 2,. . . ,m;

j, g: index of operation sequence, j, g = 1, 2,. . . ,n_i;

C_k is the completion time of M_k;

W_k is the workload of M_k.

Decision variables

x_{ijk} = 1 if machine j is selected for the operation O_{ik};
0 otherwise

c_{ik} = completion time of the operation O_{ik}

The FJSSP mathematical formulation is as follows:

$$\text{Min } f_1 = \max (C_k) \tag{1}$$

$$\text{Min } f_2 = \max (W_k) \tag{2}$$

$$\text{Min } f3 = \sum W_k \tag{3}$$

Equation (1) ensures the minimization of maximal completion time of the machines. Equation (2) ensures the minimization of maximal machine critical work load among the all the machines available. Equation (3) ensures the minimization of total work load of machines.

The weighted sum of the above three objective values are taken as the objective function:

Minimize $F= w_1 \times f_1 + w_2 \times f_2 + w_3 \times f_3$

4 RAPID MINING

Before data mining can begin, the algorithm must be determined and the data properly structured for mining. For this task, attribute-oriented induction was selected as the mining algorithm and the data was prepared accordingly.

4.1. Mining algorithm

Attribute-Oriented Induction is a set-oriented database mining method that generalizes a task-relevant subset of data, attribute-by-attribute, into a generalized relation (Cai, Cercone, & Han, 1991). This method was developed to extract characteristic rules and classifcation rules from relational databases by employing concept hierarchies into an induction process. The induction algorithm substitutes the low-level concept in a tuple with its corresponding higher-level concept, and then generalizes the relationship by eliminating identical tuples and using a threshold to control the generalization process (Han & Fu, 1996). That is, attribute by attribute, concepts which represent multiple attribute values are substituted for sets of attributes. The tuples in final relation represent rules that describe the data.

A concept hierarchy defines a sequence of mapping from a set of concepts to their higher-level correspondences. Concept hierarchies, representing necessary background knowledge, are key to the generalization process in attribute-oriented induction. They can be directly provided by users, implicitly stored in the database, or constructed automatically based on clustering behavior and data statistics. They are usually partially ordered according to a general-to-specific ordering (Han & Fu, 1996). Using concept hierarchies, the discovered rules can be represented in terms of generalized concepts which users define, and stated in a simple and explicit form.

An attribute-oriented induction process can develop two types of induction rules: learning characteristic rules (LCHR) and learning classification rules (LCLR) (Cai et al., 1991). These have similar procedures: data selection, attribute generalization, relationship simplification, and rule transformation. As their purpose is different, attribute generalization is a little different. LCHR generalizes an attribute through a relatively high-level concept, and eliminates attributes that contain a large set of distinct values without relevance to a higher-level concept. In LCLR, both the target and contrast classes may share tuples, called overlapping tuples. Because of class ambiguity for these overlapping tuples, these overlapping tuples should be marked and not generalized. Further generalization or attribute removal should rely on the unmarked tuples.

4.2. Data preparation

A data set of 10x7 FJSSP is used in this work is from Kacem et al. 40 different 10x7 sample problems were solved and 30 solution of a single problem was formed by particle swarm algorithm. 40 different problems provided a data set of 1160 examples while 30 solutions provided 870 examples set.

Table 1: 10x7 FJSSP PROBLEM

job	operation	m1	m2	m3	m4	m5	m6	m7
1	1	1	4	6	9	3	5	2
1	2	8	9	5	4	1	1	3
1	3	4	8	10	4	11	4	3
2	1	6	9	8	6	5	10	3
2	2	2	10	4	5	9	8	4
3	1	15	4	8	4	8	7	1
3	2	9	6	1	10	7	1	6
3	3	11	2	7	5	2	3	14
4	1	2	8	5	8	9	4	3
4	2	5	3	8	1	9	3	6
4	3	1	2	6	4	1	7	2
5	1	7	1	8	5	4	3	9
5	2	2	4	5	10	6	4	9
5	3	5	1	7	1	6	6	2
6	1	8	7	4	56	9	8	4
6	2	5	14	1	9	6	5	8
6	3	3	5	2	5	4	5	7
7	1	5	6	3	6	5	15	2
7	2	6	5	4	9	5	4	3
7	3	9	8	2	8	6	1	7
8	1	6	1	4	1	10	4	3
8	2	11	13	9	8	9	10	8
8	3	4	2	7	8	3	10	7
9	1	12	5	4	5	4	5	5
9	2	4	2	15	99	4	7	3
9	3	9	5	11	2	5	4	2
10	1	9	4	13	10	7	6	8
10	2	4	3	25	3	8	1	2
10	3	1	2	6	11	13	3	5

Table 2: 10x7 FJSSP PROBLEM (MINING FORMAT)

job	operation	m1	m2	m3	m4	m5	m6	m7

ISSN 2229-5518

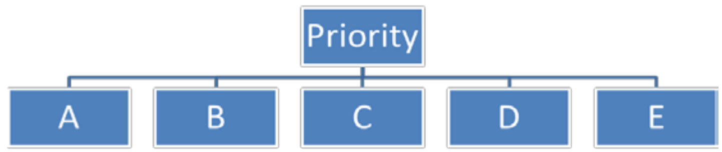
first	first	small	medium	medium	large	small	medium	small
first	second	large	large	medium	medium	small	small	small
first	third	medium	large	large	medium	large	large	small
second	first	large	large	large	medium	medium	large	small
second	second	small	large	medium	medium	large	large	medium
third	first	large	medium	large	medium	large	large	small
third	second	large	medium	small	large	large	small	medium
third	third	large	small	large	medium	small	small	large
fourth	first	small	large	medium	large	large	medium	small
fourth	second	medium	small	large	small	large	small	medium
fourth	third	small	small	medium	medium	small	large	small
fifth	first	large	small	large	medium	medium	small	large
fifth	second	small	medium	medium	large	medium	medium	large
fifth	third	medium	small	large	small	medium	medium	small
sixth	first	large	large	medium	large	large	large	medium
sixth	second	medium	large	small	large	medium	medium	large
sixth	third	small	medium	small	medium	medium	medium	large
seventh	first	medium	medium	small	medium	medium	large	small
seventh	second	medium	medium	medium	large	medium	medium	small
seventh	third	large	large	small	large	medium	small	large
eigth	first	medium	small	medium	small	large	medium	small
eigth	second	large	large	large	large	large	large	large
eigth	third	medium	medium	large	Large	small	large	large
ninth	first	large	medium	medium	medium	medium	medium	medium
ninth	second	medium	small	large	Large	medium	medium	small
ninth	third	large	medium	large	small	medium	medium	small
tenth	first	large	medium	large	large	large	medium	large
tenth	second	medium	small	large	small	large	small	small
tenth	third	small	small	medium	large	large	small	medium

The data provided by PSO is mapped into different classes for better understandings by the miner. The concept hierarchies necessary for attribute-oriented induction were developed from the operations table and not from the data to be mined. That is, it is important that the concept hierarchies be related to the problem and not the data in the solution set. The following classification hierarchies are based on the 10x7 job shop problem. Other sizes of job shop problems may require other classification approaches.

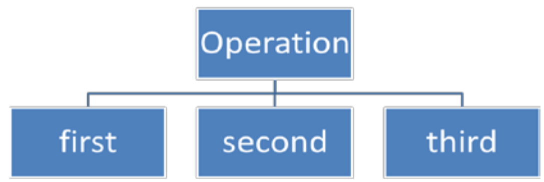
4.2.1. Priority

The mining task is to find the relationship between an operation's characteristics and its order in the PSO solution sequence. That is, we seek to predict the sequence position of an operation given its characteristics. With 29 sequence positions possible, it was decided that five abstract concepts would be substituted during rule induction. The attribute Priority is defined as a range of sequence positions in the PSO solution. Thus, the value of position is classified into one of five classes: A (1, 2, 3, 4, 5 or 6), B (7, 8, 9, 10, 11 or 12), C (13, 14, 15, 16, 17 or 18), D (19, 20, 21, 22, 23 or 24) and E (25, 26, 27, 28 and 29).

Fig 2: PRIORITY CLASSIFICATION



4.2.2. Operation



The Operation attribute is an ordinal variable representing the sequence of the operation in the job. It was decided that four classes of operation would be adequate for induction. Operation 1 and 2 was classified as “first”, operations 3,4 and 5 as “middle”, operations 6 and 7 “last”.

Fig 3: OPERATION CLASSIFICATION

4.2.3. Process

Process and Remain Process are attributes dependent on the problem domain. Process represents the time for processing for that particular operation and Remain Process represents the cumulative processing time for all subsequent operations for that job. Each of these attributes are classified into three classes: the first 1/3 as “small”, the second 1/3 as “middle”, and the third as “large”.

Fig 4: PROCESS AND REMAINING TIME CLASSIFICATION

The mining part is performed in two separate steps in hierarchical manner. Routing and scheduling is done in separate miners. Initially the machines are allocated to operations and

ExampleSet (29 examples, 4 special attributes, 9 regular attributes) View Filter (29 / 29): all

Row No.	confidence(...	confidence(...	confidence(...	prediction(s...	job	operation	1	2	3	4	5	
1	0.233	0.207	0.560	first	first	first	small	medium	medium	large	small	m
2	0.448	0.449	0.103	middle	first	second	large	large	medium	medium	small	s
3	0.208	0.204	0.588	first	first	third	medium	large	large	medium	large	la
4	0.157	0.668	0.175	middle	second	first	large	large	large	medium	medium	la
5	0.288	0.668	0.044	middle	second	second	small	large	medium	medium	large	la
6	0.148	0.313	0.539	first	third	first	large	medium	large	medium	large	la
7	0.697	0.178	0.125	last	third	second	large	medium	small	large	large	s
8	0.576	0.218	0.206	last	third	third	large	small	large	medium	medium	s
9	0.291	0.337	0.372	first	fourth	first	small	large	medium	large	large	m
10	0.520	0.119	0.361	last	fourth	second	medium	small	large	small	large	s
11	0.061	0.823	0.116	middle	fourth	third	small	small	medium	medium	small	la
12	0.383	0.321	0.296	last	fifth	first	large	small	large	medium	medium	s
13	0.161	0.187	0.652	first	fifth	second	small	medium	medium	large	medium	m
14	0.161	0.107	0.732	first	fifth	third	medium	small	large	small	medium	m
15	0.393	0.418	0.189	middle	sixth	first	large	large	medium	large	large	la
16	0.133	0.573	0.294	middle	sixth	second	medium	large	small	large	medium	m
17	0.098	0.695	0.207	middle	sixth	third	small	medium	small	medium	medium	m
18	0.040	0.589	0.371	middle	seventh	first	medium	medium	small	medium	medium	la
19	0.068	0.111	0.821	first	seventh	second	medium	medium	medium	large	medium	m

in the second step the scheduling part is obtained. The results

ExampleSet (29 examples, 6 special attributes, 6 regular attributes) View Filter (29 / 29): all

Row No.	confidence(A)	confidence(...	confidence(...	confidence(...	confidence(...	prediction(p...	job	operation	solution	process	remaining	r
1	0.674	0.265	0.001	0.059	0.000	A	first	first	first	small	small	lig
2	0.018	0.212	0.026	0.371	0.372	D	first	second	middle	small	small	lig
3	0.000	0.018	0.269	0.299	0.415	D	first	third	last	small	small	me
4	0.000	0.834	0.022	0.144	0.000	B	second	first	middle	small	small	me
5	0.000	0.565	0.030	0.202	0.203	B	second	second	first	small	small	lig
6	0.405	0.426	0.000	0.169	0.000	B	third	first	last	small	small	me
7	0.012	0.362	0.000	0.580	0.045	C	third	second	middle	small	small	lig
8	0.000	0.094	0.000	0.749	0.156	C	third	third	first	small	small	lig
9	0.394	0.481	0.002	0.123	0.000	B	fourth	first	last	small	small	lig
10	0.004	0.219	0.045	0.317	0.415	D	fourth	second	last	small	small	he
11	0.000	0.020	0.279	0.327	0.375	D	fourth	third	middle	small	small	me
12	0.548	0.377	0.000	0.075	0.000	A	fifth	first	first	small	small	he
13	0.013	0.513	0.000	0.412	0.061	B	fifth	second	first	small	small	lig
14	0.000	0.095	0.000	0.738	0.167	C	fifth	third	first	small	small	he
15	0.310	0.000	0.216	0.475	0.000	C	sixth	first	middle	small	small	me
16	0.001	0.000	0.532	0.390	0.077	E	sixth	second	middle	small	small	me
17	0.000	0.000	0.834	0.124	0.042	E	sixth	third	middle	small	small	me
18	0.879	0.000	0.020	0.101	0.000	A	seventh	first	middle	small	small	me
19	0.009	0.000	0.219	0.401	0.371	C	seventh	second	last	small	small	me

are based on probabilities and accordingly selection is made. Final solutions are present in the order shown below:

Fig 9: OPERATION PRIORITY (SCHEDULING)

In the above figures, various confidence levels along with the predictions are provided by the miner. Based on the predictions and confidence level results are extracted out of the mined data. Final solution out of extraction is shown in table 3.

Table 3: Data miner solution conversion

job	job	operation	operation	solution	solution	priority	priority
first	1	first	1	first	1	A	3
first	1	second	2	last	6	B	7
first	1	third	3	last	7	C	15
second	2	first	1	last	7	B	9
second	2	second	2	first	1	C	18
third	3	first	1	last	7	A	1
third	3	second	2	last	6	B	12
third	3	third	3	last	6	C	14
fourth	4	first	1	first	1	B	8
fourth	4	second	2	first	2	D	21
fourth	4	third	3	medium	5	E	25
fifth	5	first	1	first	2	A	2
fifth	5	second	2	first	1	C	13
fifth	5	third	3	first	2	C	17
sixth	6	first	1	medium	3	C	16
sixth	6	second	2	medium	3	D	24

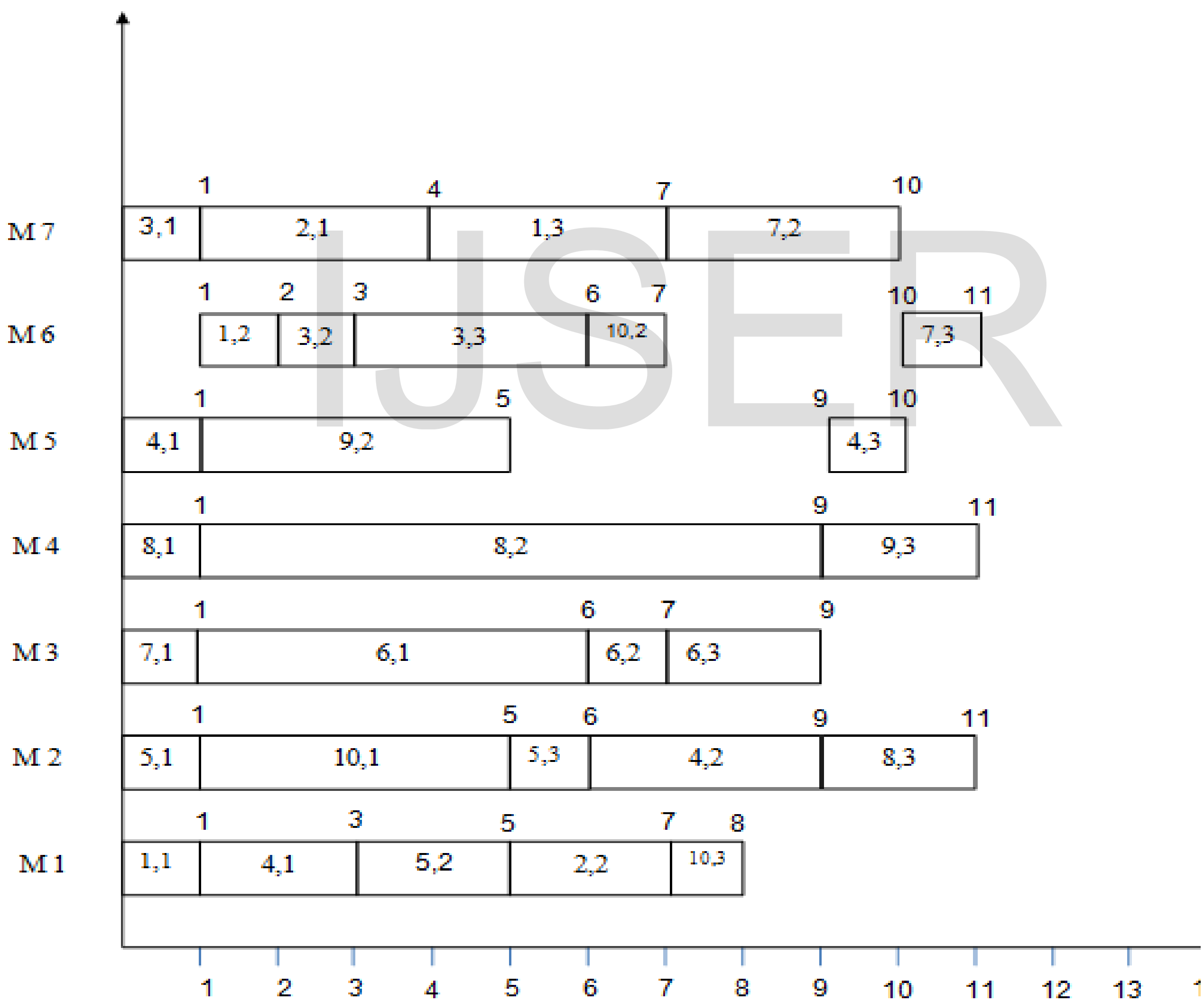
4 Results And Discussion

4.1 Computational Results

The mining procedure for FJSSP was implemented in Rapid Miner on a personal computer with Intel core 2 duo processor of 2.1 GHz and 4GB RAM. To illustrate the effectiveness and performance of the algorithm, a 10x7 problem is taken from Kacem.et.al and Weijun Xia.et.al.

The results obtained from Rapid miner are converted into routes and sequences. On basic conversion and upgradation of the mining data optimum results are achieved. The best result obtained is printed below for reference. Machine allocation and sequencing is done separately for ease of mining. Results obtained by machine allocation are fed into sequencing set for final results.

sixth	6	third	3	medium	3	E	26
seventh	7	first	1	medium	3	A	5
seventh	7	second	2	last	7	D	23
seventh	7	third	3	last	6	E	29
eighth	8	first	1	medium	4	A	4
eighth	8	second	2	medium	4	B	11
eighth	8	third	3	first	2	E	28
ninth	9	first	1	medium	5	A	6
ninth	9	second	2	medium	5	D	19
ninth	9	third	3	medium	4	E	27
tenth	10	first	1	first	2	B	10
tenth	10	second	2	last	1	D	22
tenth	10	third	3	first	6	D	20



Gantt chart for problem 10 X 7 (Makespan = 11, Max (W_k) = 11, Wtd = 66)

Table 4: Comparison of Results on Problem 10 X 7 with 29 Operations

ISSN 2229-5518

	FL+EA	MOPSO+LS	PSO+DM
MAKESPAN	16	15	11
CRITICAL LOAD	12	11	11
TOTAL WORK LOAD	60	61	66

5.2 Discussion

The computational results show that the proposed algorithm performs well when compare with the previous works on this problem. The optimal sequence obtained is represented in the Gantt chart. The obtained result is compared with the original paper (Kacem et. al.2002) and with the recently published paper (Moslehi & Mahnamn 2011)

A significant improvement in the makespan is found in the solution set, i.e. from 15 in earlier papers to 11. In this work the method is proposed for 10x7 problem but can also be implemented to other FJSSP problems

6. Conclusion

Flexible job shop scheduling is very important in both fields of combinatorial optimization and production management. Recently, multi-objective flexible job-shop scheduling problem has attracted many researchers attention. The complexity of this problem leads to the appearance of many heuristic approaches, and this work is mainly concentrated on use of Data Mining technique to solve the multi-objective flexible job shop scheduling problem. The performance of the proposed approach is evaluated in comparison with the results obtained from others work to this type of problem . The obtained computational results demonstrated the effectiveness of the proposed approach in solving the multi-objective FJSSP and a more comprehensive computational study should be made to test the efficiency of proposed solution technique. Furthermore, applying DM to other combinatorial optimization problems is also possible in further research.

7. References

1.Weijun xia and Zhiming wu (2005), An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems. Computers and Industrial Engineering. Vol.48, 409-425.

2.Haipeng Zhang, and Mitsuo Gen(2005), Multistage-based genetic algorithm for flexible job-shop scheduling problem,p.no. 223-232.

3. Xia, W. J., & Wu, Z. M. (2005). An effective hybrid optimization approach for multi objective flexible job-shop scheduling problems. Computers & Industrial Engineering, 48(2), 409–425.

4.Haoxun Chen,JurgenIhlow and Carsten Lehmann(1999), A Genetic Algorithm for flexible job-shop scheduling.

5.Guohui Zhang, Xinyu Shao , Peigen Li, Liang Gao (2008) An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem; doi:1-10.

6. Kacem, S. Hammadi, P. Borne (2002), Approach by localization and multi-objective evolutionary optimization for flexible job-shop scheduling problems, IEEE Trans.Syst. Man Cybern. C 32 (1),1–13.

7. Nozha Zribi and Pierre Borne(2005),Hybrid Genetic Algorithm for the Flexible Job-Shop Problem Under Maintenance Constraints.

8. Imed Kacem, Slim Hammadi (2002), Approach by localization and multi-objective Evolutionary optimization for flexible job-shop scheduling problems. IEEE Trancations on Systems, Man and Cybernetics- Part C; vol. 32, 1-13.

9. Imed Kacem, Slim Hammadi, (2002), Pierre borne, Pareto-Optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic. Mathematics and computers in simulation. Vol.60, 245-276.

10.Mastrolilli, M., and Gambardella, L. M. (2000). Effective neighbourhood functions for the flexible job shop problem. Journal of Scheduling, 3(1), 3–20.

11. Brandimarte, P. (1993). Routing and scheduling in a flexible job shop by taboo search. Annals of Operations Research, 41(3), 157–183.

12. Klaus Jansen, Monaldo Mastrolilli, and Roberto Solis-Oba (2000), Approximation Algorithms for Flexible Job Shop Problems. pp. 68-77.

13. Jie Gao, Mitsuo Gen, Linyan Sun, Xiaohui Zhao (2007), A hybrid of genetic algorithm and bottleneck shifting for multiobjective flexible job shop scheduling. Computers and Industrial Engineering. Vol.53, 149-162.

14. J. C. Chen & K. H. Chen & J. J. Wu & C. W. Chen,2008, A study of the flexible job shop scheduling problem with parallel machines and reentrant process , 39:344–354.

15.Moslehi G, Mahnam M (2011) A Pareto approach to multiobjective flexible job-shop scheduling problem using particle swarm optimization and local search. Int J Prod Econ 129:14–22

16. D. A. Koonce and S. C. Tsai, “Using data mining to find patterns in genetic algorithm to a job shop schedule”, Computer and industrial engineering vol. 38, pp. 361-374, 2000.