AN IMPROVE OBJECT-ORIENTED APPROACH FOR MULTI-OBJECTIVE FLEXIBLE JOB-SHOP SCHEDULING PROBLEM (FJSP)

BamaiyiSule and Ibrahim Lawal

Department of Mathematical Sciences, Abubakar
Tafawa Balewa University, Bauchi

ABSTRACT

Flexible manufacturing systems are not easy to control and it is difficult to generate controlling systems for this problem domain. Flexible job-shop scheduling problem (FJSP) is one of the instances in this domain. It is a problem which acquires the job-shop scheduling problems (JSP). FJSP has additional routing sub-problem in addition to JSP. In routing sub-problem each task is assigned to a machine out of a set of capable machines. In scheduling sub-problem, the sequence of assigned operations is obtained while optimizing the objective function(s). In this work an object-oriented (OO) approach with simulated annealing algorithm is used to simulate multi-objective FJSP. Solution approaches provided in the literature generally use two-string encoding scheme to represent this problem. However, OO analysis, design and programming methodology helps to present this problem on a single encoding scheme effectively which result in a practical integration of the problem solution to manufacturing control systems where OO paradigm is frequently used. Three parameters are considered in this paper: maximum completion time, workload of the most loaded machine and total workload of all machines which are the benchmark used to show the propose system achieve effective result.

KEYWORDS

Object-oriented manufacturing control, Object-oriented design, Multi-objective flexible job shop scheduling, simulated annealing algorithm.

1. INTRODUCTION

Flexible job shop scheduling is a scheduling mechanism that inherits the problem of job shop scheduling problem[1]. Flexible job shop scheduling has two more additional problems to that of its predecessor job shop scheduling. Job shop scheduling is complexto control because it has many routes and job can be assign to the entire available route at a time. Flexible job shop scheduling has two sub-routing problem: assigning of operation to various available machine and sequence of operation of the machine in order to obtain an optimal solution. Obtaining optimal solution is difficult to achieve with the traditional approach to optimizing scheduling problem due to its high complexity [16]. Scheduling problem has drawn the attention of many researchers to contribute in diverse ways of addressing the problem [9].

Design of a software system is an influential factor in the industry as it controls all activities and final products in the industry. The idea of object oriented (OO) programming has been on-going

DOI: 10.5121/ijcsit.2019.11508

for decades but it significance in facilitating the development of high speed and processing capabilities in developing high quality software is just realized and incorporated in recent times [29].Object oriented programming is a programming methodology that is highly structured and incorporated with strong rational and modularity capability software development [38]. OO programming methodology is one of the major approaches in software development and it is therefore, necessary for programmers to adopt this methodology and it's appropriateness in the development of a software system.

2. PROBLEM DEFINITION

Organizing and scheduling of job is an important issue in our industry today. Even-though both the organizing and scheduling serve different purposes, both are put together to accomplish a smooth operation in the system. Scheduling environment can be categorized into flow shop scheduling where all jobs passes through all the machines in the same order as they are arranged and job shop where the order of operation among jobs varies [35].

2.1 Job-Shop Scheduling Problem

In a classical job shop scheduling problem, n number of jobs J is to be processed on a number of available machines M [3]. The sequence of processing of the jobs contained some predetermined procedure which will run on any of the available machine without interruption for a particular period of time[7]. The major challenge with the job shop scheduling problem is the sequence of operation which makes it difficult to be solved [42][35].

2.2 Flexible Job Shop Scheduling

Flexible job shop scheduling is a branch of classical job shop scheduling problem which allow sequence of operation to execute on any set of a giving machine [16]. The flexible job shop scheduling problem inherit all the attribute of its predecessor with a more complex features, assigning of job to the available machines and sequence of operation of each of the job assign to the machines [42][16][28][23]. The two additional problems by the flexible job shop scheduling, makes it to be and NP-hard problem [42.

2.2.1 Single Objective Flexible Job Shop Scheduling

Many researchers have tried to optimize the performance of machine using single objective; mostly optimized objective is make span (Completion time of the job) which makes the problem clearly defined. The disadvantage of using a single objective is that is makes that particular used objective to look more superior to other necessary objectives to be considered [36][31].

2.2.2 Multi Objective Flexible Job Shop Scheduling

Multi objective flexible job shop scheduling uses several objectives to optimize the performance of the machine. This method makes other objectives not to be superior to others and it also gives good compromise solution for the decision makers to select one out of the several others [36].

2.3 Method for Solving Optimizations Problem

There are several methods used for solving optimization problem, some of the methods mentioned by [3] are as follows:

2.3.1 Exact Method

The exact method is used to solve job shop scheduling problem. They provide optimal solution only for particular instance of a giving solution. Some of the exact method algorithms are not consistently successful in solving job shop scheduling. Examples of such algorithm are dynamic programming and branch and bound.

2.3.2 Approximated Method

In the approximated method algorithm some quality of result are prepared and presented which will be used to ascertain the quality of the expected result. Thismethod provides an insight to the distance of the approximated result to the optimum result.

2.3.3 Heuristic Method

Heuristic method provide a good solution to any instance of a given problem that work well with different problem size, less computational time and easily combined with other method to obtained optimum solution. The major setback with the heuristic method is that the more the size of the problem increases the more it fails to offer optimum solution.

2.3.4 Meta heuristic Method

Meta heuristic method is an improved heuristic method can be applied in solving various combinatorial and NP-hard problems. It's broadly described as constructive approach where each step takes presiding output as input and constructs new sequence of output.

Fundamental Properties of Meta-heuristics as stated by [4]. are mentioned below:

Meta-heuristics are strategies that "guide" the search process.

The goal is to efficiently explore the search space in order to find (near) optimal solutions.

Techniques which constitute meta-heuristic algorithms range from simple local search procedures to complex learning processes.

Meta-heuristic algorithms are approximate and usually non-deterministic.

They may incorporate mechanisms to avoid getting trapped in confined areas of the search space.

The basic concepts of meta-heuristics permit an abstract level description.

Meta-heuristics are not problem-specific.

Meta-heuristics may make use of domain-specific knowledge in the form of heuristics that are controlled by the upper level strategy.

Today's more advanced meta-heuristics use search experience (embodied in some form of memory) to guide the search.

Some of the most notable groups of meta-heuristics approaches using to solve combinatorial optimization problems are: simulated annealing, tabu search, evolutionary algorithms, ant colony optimization and particle swarm optimization

There are two types of global optimization algorithm: deterministic and random algorithms[25]. The deterministic algorithm uses the deterministic search strategy while the random algorithm uses a technique that introduce random factors in the appropriate search strategy and the result it generates is probabilistically in nature. [25].

Simulated annealing (SA) is a random algorithm which employ the random-search technique and its idea is based on heating solid using a very high temperature to melt first, and allow it cool slowly solidified into a good crystal as the temperature decreases. When heating the fixed, the inside of the solid particles can increase the internal energy with the increase of internal temperature. When internal energy achieves maximum, the arrangement state of the particle into a liquid disordered. This process is called smelting. When cooling, Particle solidified into a solid crystalline state with the decrease of the temperature. The particle is orderly and solidified into a solid crystalline state. This process is called annealing. When the internal energy is reduced to the minimum, and finally reached the ground state at normal temperature which exploits an analogy between the ways in which a metal cools and freezes into a minimum energy crystalline structure (the annealing process) and the search for a minimum in a more general system; it forms the basis of an optimization technique for combinatorial and other problems.

Simulated annealing was developed in 1983 to deal with highly nonlinear problems as well as to address discrete and continuous optimization problems.

3. RELATED WORKS

Genetic algorithm (GA) was proposed by [13].Tooptimize the make span of a flexible job shop scheduling problem. In this research, the methodology adopted was to allow all the jobs to arrive at a time t=0 so that the starting and the completion time of the job can be monitored in order to obtain a minimized make span. In the methodology adopted in the paper, a fitness test was conducted on all the chromosomes generated during the machine operation and the chromosome with small completion time is considered to be the one with an optimized and good performance. The result in the study shows that GA provides a better optimization of the job make span. However, the challenge with GA is that an optimal solution is difficult to obtain. More so, [31]. "A Hybrid Multi Objective Algorithm for Flexible Job Shop Scheduling" proposed a Pareto approach to solve the multi objective flexible job shop scheduling problems. The objectives considered are to minimize the overall completion time (makespan) and total weighted tardiness (TWT). An effective simulated annealing algorithm based on the proposed approach is presented to solve multi objective flexible job shop scheduling problem. An external memory of non-dominated solutions is considered to save and update the non-dominated solutions during the solution process and the algorithm was run on a PC that has a Pentium-IV 1.80 GHz processor, with 512 Mb RAM. In the experimental result, numerical experiments show that the proposed algorithm is capable to obtain the solution near the optimal. Moreover, the proposed algorithm can obtain all of Pareto solution in a small time. Therefore, the proposed algorithm is useful in multi objective flexible job shop scheduling problems and can be applied easily in real factory conditions and for large size problems. The problem associated with this method is that only two well-known objectives are used for the multi objective flexible job shop scheduling problems, so a review on another objectives and methods in this field can be supposed as further research.

[15] in their research "Solving Job Shop Scheduling Problem Using an Ant Colony Algorithm" proposed and describes the implementation of an ant colony algorithm (ACA), applied to a combinatorial optimization problem called job shop scheduling problem (JSSP). They apply an ant colony algorithm (ACA) to solve the JSSP with the objective of minimizing the maximum completion time, or makespan. The proposed algorithm is based on model designed by [11] for the permutation flow shop problem. Also, computer simulations on a set of benchmark problems were conducted to assess the merit of the proposed algorithm compared to some other heuristics in the literature. A novel mechanism is employed in initializing the pheromone trails based on an initial sequence. Moreover, the pheromone trail intensities are limited between lower and upper bounds which change dynamically. The result of the algorithm was coded in Visual C++ and all test runs carried out on a 2.0 GHz Intel Core 2 Duo Processor with 2 GB memory. Their experimental results show that the proposed algorithm is competitive when compared with the best known solutions in the literature. But the problem associated with this research is that they only minimized one objective function for the job shop problem, so a review of other objective functions using an ant colony algorithm (ACA) in this field can be supposed as further research [2].FurthermoreIntheir study "Performance of Genetic Algorithms for Solving Flexible Job-Shop Scheduling Problem" implemented and compared two approaches i.e. Jobs Sequencing List (JSL) Oriented Genetic Algorithm and Operations Machines Coding (OMC) Oriented Genetic Algorithm. In their work each approach has its own coding, evaluation function, crossovers and mutations applicable in Job-Shop scheduling problem to minimize the make span, the workload of the most loaded machine and the total workload of the machines which they considered as their objective functions.

They consider the following

Constraints:

1. For each job, the order of operations is fixed.
2. A machine can only execute one operation at a given time: it becomes available to others.

Hypothesis:

1. All machines are available at t = 0.
2. All jobs can be started at t = 0.

Their objective is to compare the two genetic algorithms with different chromosome representations, different evaluation function, different crossovers and mutations and find out which algorithm minimizes better depending upon the following criteria: makespan, the workload of the most loaded machine and the total workload of the machines. From their research their results show that (OMC) Oriented GA is good when makespan is only considered. When all three objectives makespan, workload of the most loaded machine and total workload of the machine are considered, (JSL) Oriented GA outperformed (OMC) Oriented GA in terms of minimizing the objectives but converged late. (I.e. problem with their work is the late convergence of the JSL). Therefore (JSL) Oriented GA has been found to be the best out of two approaches to minimize the objectives.

[11]in their study "Multi objective Flexible Job Shop Scheduling Optimization Using BFOA" proposes an adaptive bacterial foraging optimization algorithm (ABFOA) for solving

multiobjective flexible job-shop scheduling problem (FJSP). Bacterial foraging optimization algorithm (BFOA) is a new bio-inspired evolutionary computational technique, which is inspired by foraging behaviour of E. coli bacterium present in the human intestine. In their study, to make this algorithm suitable to combinatorial nature of FJSP and eliminate the complexities of original BFOA, some modifications are proposed, which enhance the convergence characteristics of the algorithm. The objective of their research is to minimize makespan, total workload of machines and the maximal workload. They consider 9 different Hypotheses and adopted Pareto-optimal concept to solve the Multi objective Flexible Job Shop Scheduling Problem (MOFJSP). From their results, the solution obtained by the proposed algorithm reached the optimal solutions as obtained by other algorithms of the literature. Therefore, the proposed approach is flexible enough to be used for both total-FJSP and Partial-FJSP. The problem associated with this study is high computation time because it takes more time for computation to complete. For further research, working on using adaptive BFOA for solving more realistic flexible job shop scheduling problems can be considered as a good research area. In the same vein, [3], used an improved genetic algorithms (IGA)to provide solution to multi objective flexible job shop scheduling problem. In their work, two objective functionsmake span and processing cost were used to minimize the operation of the machine. The methodology adopted in the work was the collection of both standard genetic algorithm (SGA) and Genetic algorithm (GA) so as to have a mix algorithm to address the MOFJSP. The experimental result shows that, the improved genetic algorithm (IGA) performs better than other mixed algorithm used in the literature and it also improve performance to some certain extend.

The several concepts used by researchers in addressing FJSP wasanalysed by [38]. The work pointed out the most used algorithms by researchers in providing solution to FJSP. Two approachesare used in the work, mathematical model used for addressing small problems and meta-heuristic approach for addressing both small and large problems. Most of the algorithms used in the literature provide solutions based on pareto approached which provide a set of solutions and the work proposed the use of approach that can provide single optimal solution. Also, a Memetic Algorithms was proposed to solve a FJSP problem. The performance of the Memetic algorithm was compared with some other algorithms like the state-of-the-art algorithm and non-sorting genetic algorithm. The performance of the Memetic algorithms stands to outperform the other selected algorithms. In their study, three objectives i.e.makespan, total workload and critical workload was used to analyze the performance of the machine. Even though, the Memetic algorithms standout ahead of the other selected algorithms, yet it performance is still arguable as it is subjected to improve in performance[41].

[8].Proposedan Elitist selection genetic algorithm to address multi resource flexible job shop scheduling problem. In their contribution, four resources were considered, Machine, Warehouse, Vehicles and Detection equipment to maximize the makespan of the machine. The result of Elitist selection genetic algorithm shows that using the four resource, MRFJSP shows a better performance as compared to the single flexible job shop scheduling problem.

4. MULTI-OBJECTIVE FLEXIBLE JOB-SHOP

In trying to provide solution to flexible job-shop scheduling problem, several other approaches have been used to provide solution based on a double string encoding scheme. This research proposes the use of object oriented programming (OOP) methodology using simulated annealing (SA) algorithm to provide a solution to flexible job shop scheduling problem with single string encoding scheme.

Related Job and machine data could be processed and it information recorded to a DBMS based OO programming methodology. This data could be represented in objects and could be used on scheduling and data record/retrieval purposes, representing some basic attributes and behaviours of a typical job and machine class in a manufacturing information system.

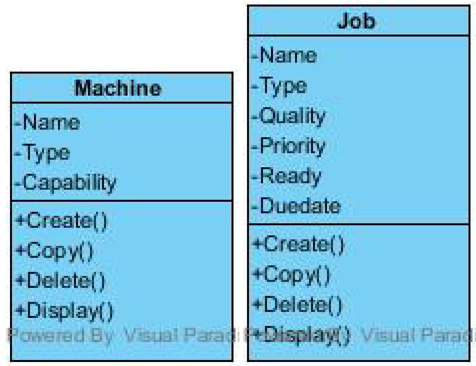

Fig.1. Job and machine class definition

4.1 Problem formulation

There two problems to address when dealing with flexible job shop scheduling problem, assigning of jobs machine and sequence of operation of machine on each job. The problem of job shop scheduling as defined by [4] is as follows:

- There are n jobs, indexed by i, and these jobs are independent on each other.
- Each i has an operating sequence, denoted by ji (precedence constraint). Ji denote the i-th job to be assign to the machine
- Each operating sequence is ordered set of operation Oij for J=1,…..ni.
- There are m machines indexed by k (the k-th machine is denoted my mk).
- For each operation Oij, there is a set of machines capable of executing it. The set is denoted by Mi,j, Mi,j ϵ {1,…m} routing constraint if it does exist.
- The processing time for an operation Oi,j on machine k is predefined and denoted by byti,j,k.

4.1.1 Assumptions

1. Each operation cannot be interrupted during its performance (Non-preemptive).
2. Each machine can perform at most one operation at a giving time.
3. Each machine becomes available to another operation once the operation which are assigned is completed
4. All machine are available at time t = 0.
5. All jobs can be started at time t = 0.
6. Precedence constraints of the operation in a job can be defined for any pair of operation.
7. Setting up time of machines and move time between operations are negligible.
8. Machines are independent from each other.
9. There are no precedence constraints among operations of different jobs.
10. Release time or due dates are not specified [4].

4.2 Research Framework

4.2.1 Multi Objective Flexible Job Shop Scheduling Analysis

In the review of literatures so many approaches were applied to solve flexible job shop scheduling problem. Most of the researcher's uses single objective to address the problem while others use multi objectives approaches but all provide solution based on double string encoding scheme. This work tries to imbibe the idea of OO programming with multi objectives to address the flexible job shop scheduling problem.

4.2.2 Multi Objectives Flexible Job Shop Scheduling Design

Objected oriented programming is a programming approach that was recently adopted in manufacturing industries. The OO programming breaks down problem into a collection of classes and objects. The concept of OO programming is very good because it allows integration of different algorithms to solve a problem and also support the addition of a system without building it from a scratch using inheritance capabilities. This research uses OO programming approach not just to provide good result but also to provide a better solution to scheduling problems.

4.2.3 Design Tools

For the sake of this research work, we used java netbeans to simulate the scheduling problem on intel core CPU, 4GB Ram and 2.20GHz. The utilization of the machine was used to determine which of the machine is more loaded with jobs and the machine that execute fewer jobs.

5. EVALUATION AND VALIDATION

Considering several approaches used in the literature by different researchers the OO programming approach is assumed to be the current approach and provide a better solution to addressing the scheduling problem. It also breached the gap between software developers and algorithms designers which was considered as a formidable task using other approaches.

Table1. 4X5 Problems

		M1	M2	M3	M4	M5
Job1	$O_{1,1}$	2	5	4	1	2
	$O_{1,2}$	5	4	5	7	5
	$O_{1,3}$	4	5	5	4	5
Job2	$O_{2,1}$	2	5	4	7	8
	$O_{2,2}$	5	6	9	8	5
	$O_{2,3}$	4	5	4	54	5
	$O_{2,4}$	4	5	2	1	5
Job3	$O_{3,1}$	1	5	2	4	12
	$O_{3,2}$	5	1	2	1	2

Table 2. 8x8Problems

		M1	M2	M3	M4	M5	M6	M7	M8
Job1	**01,2**	**5**	**3**	**5**	**3**	**3**	**-**	**10**	**9**
		10	**-**	**5**	**8**	**3**	**9**	**9**	**6**
		-	**10**	**-**	**5**	**6**	**2**	**4**	**5**
Job2		**5**	**7**	**3**	**9**	**8**	**-**	**9**	**-**
		-	**8**	**5**	**2**	**6**	**7**	**10**	**9**
		-	**10**	**-**	**5**	**6**	**4**	**1**	**7**
Job3		**10**	**-**	**-**	**7**	**6**	**5**	**2**	**4**
		-	**10**	**6**	**4**	**8**	**9**	**10**	**-**
		1	**4**	**5**	**6**	**-**	**10**	**-**	**7**
Job4		**3**	**1**	**6**	**5**	**9**	**7**	**8**	**4**
		12	**11**	**7**	**8**	**10**	**5**	**6**	**9**
		4	**6**	**2**	**10**	**3**	**9**	**5**	**7**
Job5		**3**	**6**	**7**	**8**	**9**	**-**	**10**	**-**
		10	**-**	**7**	**4**	**9**	**8**	**6**	**-**
		11	**9**	**-**	**6**	**7**	**5**	**3**	**6**
Job6		**6**	**7**	**1**	**4**	**6**	**9**	**-**	**10**
		11	**-**	**9**	**9**	**9**	**7**	**6**	**4**
		10	**5**	**9**	**10**	**11**	**-**	**10**	**-**
Job7		**5**	**4**	**2**	**6**	**7**	**-**	**10**	**-**
		-	**9**	**-**	**9**	**11**	**9**	**10**	**5**
		-	**8**	**9**	**3**	**8**	**6**	**-**	**10**
Job8		**2**	**8**	**5**	**9**	**-**	**4**	**-**	**10**
		9	**9**	**-**	**8**	**5**	**6**	**7**	**1**
		9	**-**	**3**	**7**	**1**	**5**	**6**	**1**

Table 3. 10x10 Problems

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
Job1	1	4	6	9	3	5	2	8	9	5
	4	1	1	3	4	8	10	4	11	4
	3	2	5	1	5	6	9	5	10	3
Job2	2	10	4	5	9	8	4	15	8	4
	4	8	7	1	9	6	1	10	7	1
	6	11	2	7	5	3	5	14	9	2
Job3	8	5	8	9	4	3	5	3	8	1
	9	3	6	1	2	6	4	1	7	2
	7	1	8	5	4	9	1	2	3	4
Job4	5	10	6	4	9	5	1	7	1	6
	4	2	3	8	7	4	6	9	8	4
	7	3	12	1	6	5	8	3	5	2
Job5	7	10	4	5	6	3	5	15	2	6
	5	6	3	9	8	2	8	6	1	7
	6	1	4	1	10	4	3	11	13	9

Job6	**8**	**9**	**10**	**8**	**4**	**2**	**7**	**8**	**3**	**10**
	7	3	12	5	4	3	6	9	2	15
	4	7	3	6	3	4	1	5	1	11
Job7	1	7	8	3	4	9	4	13	10	7
	3	8	1	2	3	6	11	2	13	3
	5	4	2	1	2	1	8	14	5	7
Job8	5	7	11	3	2	9	8	5	12	8
	8	3	10	7	5	13	4	6	8	4
	6	2	13	5	4	3	5	7	9	5
Job9	3	9	1	3	8	1	6	7	5	4
	4	6	2	5	7	3	1	9	6	7
	8	5	4	8	6	1	2	3	10	12
Job10	4	3	1	6	7	1	2	6	20	6
	3	1	8	1	9	4	1	4	17	15
	9	2	4	2	3	5	2	4	10	23

Table 4. 15X10 Problems

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
Job1	**1**	**4**	**6**	**9**	**3**	**5**	**2**	**8**	**9**	**4**
	1	**1**	**3**	**4**	**8**	**10**	**4**	**11**	**4**	**3**
	2	**5**	**1**	**5**	**6**	**9**	**5**	**10**	**3**	**2**
	10	**4**	**5**	**9**	**8**	**4**	**15**	**8**	**4**	**4**
Job2	**4**	**8**	**7**	**1**	**9**	**6**	**1**	**10**	**7**	**1**
	6	**11**	**2**	**7**	**5**	**3**	**5**	**14**	**9**	**2**
	8	**5**	**8**	**9**	**4**	**3**	**5**	**3**	**8**	**1**
	9	**3**	**6**	**1**	**2**	**6**	**4**	**1**	**7**	**2**
Job3	**7**	**1**	**8**	**5**	**4**	**9**	**1**	**2**	**3**	**4**
	5	**10**	**6**	**4**	**9**	**5**	**1**	**7**	**1**	**6**
	4	**2**	**3**	**8**	**7**	**4**	**6**	**9**	**8**	**4**
	7	**3**	**12**	**1**	**6**	**5**	**8**	**3**	**5**	**2**
Job4	**6**	**2**	**5**	**4**	**1**	**2**	**3**	**6**	**5**	**4**
	8	**5**	**7**	**4**	**1**	**2**	**36**	**5**	**8**	**5**
	9	**6**	**2**	**4**	**5**	**1**	**3**	**6**	**5**	**2**
	11	**4**	**5**	**6**	**2**	**7**	**5**	**4**	**2**	**1**
Job5	**6**	**9**	**2**	**3**	**5**	**8**	**7**	**4**	**1**	**2**
	5	**4**	**6**	**3**	**5**	**2**	**28**	**7**	**4**	**5**
	6	**2**	**4**	**3**	**6**	**5**	**2**	**4**	**7**	**9**
	6	**5**	**4**	**2**	**3**	**2**	**5**	**4**	**7**	**5**
Job6	**4**	**1**	**3**	**2**	**6**	**9**	**8**	**5**	**4**	**2**
	1	**3**	**6**	**5**	**4**	**7**	**5**	**4**	**6**	**5**
Job7	**1**	**4**	**2**	**5**	**3**	**6**	**9**	**8**	**5**	**4**
	2	**1**	**4**	**5**	**2**	**3**	**5**	**4**	**2**	**5**
Job8	**2**	**3**	**6**	**2**	**5**	**4**	**1**	**5**	**8**	**7**
	4	**5**	**6**	**2**	**3**	**5**	**4**	**1**	**2**	**5**
	3	**5**	**4**	**2**	**5**	**49**	**8**	**5**	**4**	**5**
	1	**2**	**36**	**5**	**2**	**3**	**6**	**4**	**11**	**2**
Job9	**6**	**3**	**2**	**22**	**44**	**11**	**10**	**23**	**5**	**1**
	2	**3**	**2**	**12**	**15**	**10**	**12**	**14**	**18**	**16**
	20	**17**	**12**	**5**	**9**	**6**	**4**	**7**	**5**	**6**

Job10	5	8	7	4	56	3	2	5	4	1
	2	5	6	9	8	5	4	2	5	4
	6	3	2	5	4	7	4	5	2	1
	3	2	5	6	5	8	7	4	5	2
Job11	1	2	3	6	5	2	1	4	2	1
	2	3	6	3	2	1	4	10	12	1
	3	6	2	5	8	4	6	3	2	5
	4	1	45	6	2	4	1	25	2	4
Job12	9	8	5	6	3	6	5	2	4	2
	5	8	9	5	4	75	63	6	5	21
	12	5	4	6	3	2	5	4	2	5
	8	7	9	5	6	3	2	5	8	4
Job13	4	2	5	6	8	5	6	4	6	2
	3	5	4	7	5	8	6	6	3	2
	5	4	5	8	5	4	6	5	4	2
	3	2	5	6	5	4	8	5	6	4
Job14	2	3	5	4	6	5	4	85	4	5
	6	2	4	5	8	6	5	4	2	6
	3	25	4	8	5	6	3	2	5	4
	8	5	6	4	2	3	6	8	5	4
Job15	2	5	6	8	5	6	3	2	5	4
	5	6	2	5	4	2	5	3	2	5
	4	5	2	3	5	2	8	4	7	5
	6	2	11	14	2	3	6	5	4	8

5.1 Experimented Results

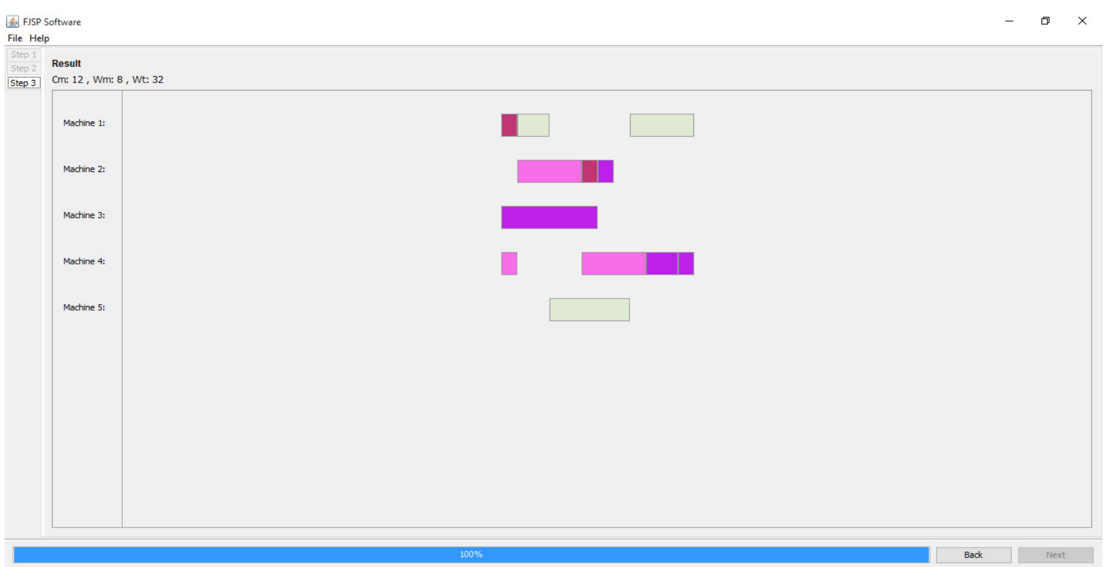

Fig.2. Schedule for problem set 4X5

Fig.3. Schedule for problem set 8X8

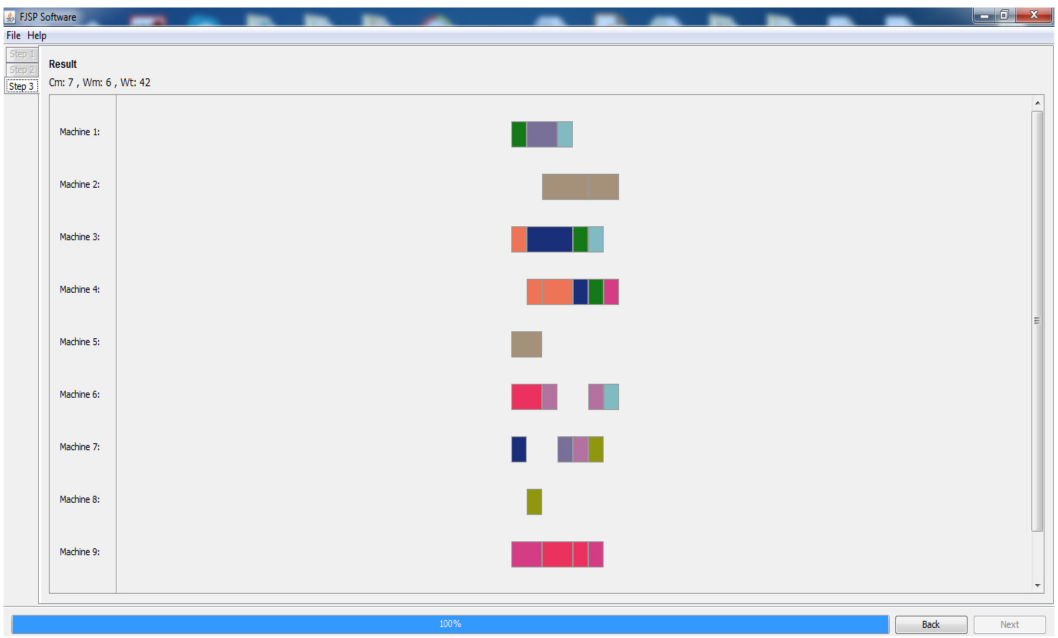

Fig.4. Schedule for problem set 10X10

Fig.5. Schedule for problem set 15X10

5.2 Comparison of Algorithms

The obtain result will be compare using different algorithm as listed below:

i. AL: Kacem et al. (2002a),(2002b)
ii. PSO+SA: Xia and wu (2005).
iii. MOGA: Saad et al. (2008).
iv. hPSO: Shao et al. (2013).
v. hGA: Gao et al. (2007).

In the table of comparison, we used F1,F2 and F3 to represent the completion time (CM), Maximum workload of the machine (WM) and the Total workload of all the machines (WT).

Table 5: Comparison of the algorithm performance for 8x8 problems set

AL			**PSO+SA**			**MOGA**			**hPSO**			**hGA**			**SA+ OOapproach**		
F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3
16	15	17	15	12	17	16	13	75	14	12	77	15	12	75	16	13	73
			16	13	73				15	12	75				15	12	75
									16	11	77						
									16	13	73						

Table 6: Comparison of the algorithm performance for 10x10 problems set

AL			**PSO+SA**			**MOGA**			**hPSO**			**hGA**			**SA+ OOapproach**		
F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3
7	5	45	7	6	44	7	5	44	7	5	43	7	5	43	7	5	43
8	5	42							7	6	42				7	6	42
8	7	41							8	5	42				7	7	42
									8	7	41						

Table 7: Comparison of the algorithm performance for 15x10 problems set

AL			**PSO+SA**			**MOGA**			**hPSO**			**hGA**			**SA+ OOapproach**		
F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3
23	11	95	12	11	91	23	11	99	11	11	91	11	11	91	12	11	92
24	11	91							12	10	93						
									11	10	95						

5.4 Discussion of result

The problem set 4X5 is given in Table.1. And best schedule found for this problem set is given in figure.1. Solution found for this problem is F1=12, F2=8 and F3=32. The best solution found for this problem set in the literature is F1=12, F2=8 and F3=32, that is the same with the solution of the proposed algorithm.

The problem set 8X8 is given in Table.2 and best schedule found for this problem set is given in figure.2. Solution found for this problem in the first run is F1=16, F2=13 and F3=73 and for the second run is F1=15, F2=12, F3=75, which is the same with the solution of the proposed algorithm.

The problem set 10X10 is given in Table.3 and best schedule found for this problem set is given in figure.3. Solution found for this problem in the first run is F1=7, F2=5 and F3=43, for the second run is F1=7, F2=6, and F3=42 and for the third run is F1=7, F2=7, and F3=42, which is the same with the solution of the proposed algorithm.

The problem set 4X5 is given in Table.4 and best schedule found for this problem set is given in figure.4. Solution found for this problem is F1=12, F2=11 and F3=91. This is the same with the solution of the proposed algorithm.

6. CONCLUSION

This work proposes an OO representation for FJSP using class diagram that reduces the problem encoding to a single data structure where hierarchical data structure is used to show operational objects of FJSP. Main advantage of the proposed approach is its capability of reducing the number of data structures used during problem solving process and increasing adaptability of algorithms to real manufacturing control systems. The proposed approach can also directly produce feasible solutions effectively. However, the number of system elements is increasing while making the system OO approach. This characteristic of the approach could be considered as a limitation of this present work. Level of data decoupling plays a crucial role in OO system design because number of different objects in the system might increase system complexity. System designers should keep the level of decoupling in such circumstances.

Experimental results shows that the proposed OO approach is able to use single algorithm to find non dominated solution to most of the problem sets, in contrary to what previous studies which use hybrid algorithm to find their solution to multi-objective FJSP. Considering the simulation results obtained, the OO approach performs better in minimizing the following objective functions: Completion time, Maximum workload and the total workload of the most loaded machine.

6.1 Future Work

Multi-objective FJSP problem is designed and solved by using OO approach and simulated annealing algorithm in this work, it is recommended to consider further work in line with this idea, might also consider using it in other algorithm, in other to improve performance of Job Shop Problems.

REFERENCE

[1] Abir, B., H., Mohamed, H., Marie, J., H., Pierre, L., (2010). Discrepancy search for the flexible job shop scheduling. Computers & operation research / Computers and operation research, Elseveir, 2010, 37 (12), p. 2192-2201.

[2] Amit Kumar and Rajnesh Singh (2011), “Performance of Genetic Algorithm for Solving Flexible Job-Shop Scheduling Problem.”International Journal of Information Technology and Knowledge Management.VOl. 4, No. 1, pp. 105-108.

[3] Anshulika, L., A., Bewoor, (2015). Application of genetic algorithm on job shop scheduling problem to minimizemakespan.International journal of science and research (IJSR).Vol. 5.No. 6.Pp 2319-7064.

[4] Arash, M., L., Kamyar, S., L., Mahdi, H., (2010). Solving flexible job shop scheduling with multi objective approach.International journal of industrial engineering and production research.Vol.21, N0. 4. Pp.197-209

.

[5] Baykasoglu, A., Ozbakir, L., &Sonmez, A. I. (2004). Using multiple objective tabu search and grammars to model and solve multi-objective flexible job-shop scheduling problems. Journal of Intelligent Manufacturing, 15(6), 777–785.

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

[7] Binay, K., S., (2013). Obtain makespan of job shop scheduling operation.

[8] Chen, H., Ihlow, J., & Lehmann, C. (1999). A genetic algorithm for flexible job-shop scheduling.In Proceedings of IEEE international conference on robotics and automation (pp.1120–1125).

[9] Deb, K. (2001). Multi-objective optimization using evolutionary algorithms. Chichester, UK: Wiley.

[10] Dusan, N. Sormaz, JaikumarArumugam, Ramachandra S Harihara, Chintankumar Patel NarenderNeerukonda(2010). Integration of product design, process planning, scheduling and FMS control using XML data representation. Robotics and Computer-Integrated Manufacturing, 26: 583-495.

[11] D. Nagendra, A.K. Sood (2015), "Multi Objective Flexible Job Shop Scheduling Optimization Using BFOA." International Journal of Emerging Technology and Advanced Engineering. Vol. 5, Issue 5, www.ijetae.com, ISSN: 2250-2459.

[12] Garey, M. R., Johnson, D. S., &Sethi, R. (1976). The complexity of flow shop and jobshop scheduling. Mathematics of Operations Research, 1, 117–129.

[13] Gaurav Kumar, Trilok Singh Bisoniya (2015), "Flexible job shop scheduling operation using genetic algorithm." International Journal of Innovation in Engineering and Technology (IJIET). Vol. 5 ISSN: 2319-1058

[14] Glover, F., & Laguna, M. (1997). Tabu search. Kluwer Academic Publishers.

[15] HabibehNazif (2015), "Solving Job Shop Scheduling Problem Using an Ant Colony Algorithm." Journal of Asian Scientific Research. Vol. 5(5):261-268. ISSN (e): 2226-1331/ISSN (p): 2226-5724.

[16] Hongbo Liu, Ajith Abraham, Zuwen Wang, "A Multi-swarm Approach to Multi-objective Flexible Job- shop Scheduling Problems", School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China, Fundamental Informaticae,IOS Press, 2009.

[17] Hsu, T., Dupas, R., Jolly, D., &Goncalves, G. (2002). Evaluation of mutation heuristics for the solving of multi-objective flexible job shop by an evolutionary algorithm.In Proceedings of the 2002 IEEE international conference on systems, man and cybernetics (Vol. 5, pp. 655–660).

[18] Hurink, E., Jurisch, B., &Thole, M. (1994). Tabu search for the job shop scheduling problem with multi-purpose machine. Operations Research Spektrum, 15, 205–215.

[19] Kacem, I., Hammadi, S., & Borne, P. (2002a). Approach by localization and multi-objective evolutionary optimization for flexible job-shop scheduling problems.IEEE transactions on systems, man, and cybernetics, Part C, 32(1), 1–13.

[20] Kacem, I., Hammadi, S., & Borne, P. (2002b). Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic. Mathematics and Computers in Simulation, 60, 245–276.

[21] Kennedy, J. (1997). Particle swarm: social adaptation of knowledge. In Proceedings of the 1997 IEEE international conference on evolutionary computation (pp. 303–308).USA: Indianapolis.

[22] Kennedy, J., &Eberhart, R. (1995). Particle swarm optimization.In Proceedings of the 1995 IEEE international conference on neural network (Vol. IV, No. 4, pp. 1942– 1948).

[23] Kong, L., Li, T., Wang, K., Zhu, H., Takano, M., Yu, B., (2015). An improved shuffled frog-leaping algorithms for flexible job shop scheduling.www.mdpi.com/journal/algorithms.

[24] Lawler, E. l., Lenstra, J. K., RinnooyKan, A. H. G., &Shmoys, D. B. (1993). Sequencing and scheduling: Algorithms and complexity. In S. C. Graves et al. (Eds.), Logistics of production and inventory (pp. 445–522). Amsterdam: North Holland.

[25] L. Ba, Y. Li, M.S Yang, X.Q Gao, Y. Liu (2016), "Modeling and Simulation Of A Multi-Resource Flexible Job Shop Scheduling" int j simul model, Vol. 15, ISSN: 1726-4529.

[26] Mastrolilli, M., & Gambardella, L. M. (2000). Effective neighborhood functions for the flexible job shop problem. Journal of Scheduling, 3(1), 3–20.

[27] Mesghouni, K., Hammadi, S., & Borne, P. (1997). Evolution programs for job-shop scheduling. In Proceedings of the 1997 IEEE international conference on systems, man and cybernetics (Vol. 1, pp. 720–725).

[28] Mohsen, Z., (2014). An efficient heuristic algorithm for flexible job shop scheduling with maintenance constraint.Journal of applied mathematics and science (MathsSJ), Vol.1, 1.

[29] Moreno Reyna, A., Gomez Ortega, A., Siera Romero. N., Armando Diaz, D., Fernandez Murillo, S. E, Felix Zarate, G. A., Narvaez Granados, S. L (2012) object oriented programming as an alternative to industrial control, New York: IEEE.

[30] M. Lan, T. Xu and L peng (2010), "Solving Flexible Multi-Objective JSP Problem Using An Improved Genetic Algorithm" journal of software, Vol.5, No. 10.

[31] ParvizFattahi (2009), "A hybrid multi objective algorithm for flexible job shop scheduling." World Academy of Science, Engineering and Technology.Vol. 26.

[32] P. Wojakowski, D. Warzolek (2013), "Research Study of State Of The Art Algorithms for Flexible Job Shop Scheduling Problem"

[33] Parsopoulos, K. E., &Vrahatis, M. N. (2002). Recent approaches to global optimization problems through particle swarm optimization. Natural Computing, 1(2-3), 235–306.

[34] Rigao, C. (2004). Tardiness minimization in a flexible job shop: A tabu search approach. Journal of Intelligent Manufacturing, 15(1), 103–115.

[35] Shantanu, K., Zanwar, D., R., (2013). Scheduling in job process industry.Journal of mechanical and civil engineering (IOSR-JMCE). ISSN: 2278-1684. Vol. 5. N0.1. pp 01-17.

[36] Soad, M., Bestma C., F., (2015). Multi objective flexible job shop scheduling using a modified invasive weed optimization. International Journal on soft computing (IJSC) Vol.6 No1.

[37] Sonmez, A. I., &Baykasoglu, A. (1998). A new dynamic programming formulation of (n * m) flow shop sequencing problems with due dates. International Journal of Production Research, 36, 2269–2283.

[38] Shi-xin, L, Meng-guang, W, (2000). An object-oriented methodology for solving the RCPSPs with heuristic and meta-heuristics.Production Planning & control. 11(5), 434-442. doi: 10.1080/09537280050051933.

[39] T. Ning, C Guo, R. Chen and H. Jin (2016), "A Novel Hybrid Method for Solving Flexible Job Shop Scheduling Problem."The Open Cybernetic and Systemic Journal, DOI: 10.2174/1874110X01610010013.

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

[41] Y. Yuan, H. Xu (2013), "A Memetic Algorithm for the Multi-Objective Flexible Job Shop Scheduling Problem."

[42] Zohreh, D., Mohammad-R, A., (2014). A novel quantum immune algorithm for multi objective flexible job shop scheduling. International journal of artificial intelligence and application for smart devices.Vol.2, No.1, pp. 57-72.