Advances and Applications in Mathematical Sciences
Volume 17, Issue 1, November 2017, Pages 231-249

A PRIORITY BASED HYBRID EVOLUTIONARY ALGORITHM APPROACH TO MULTI-OBJECTIVE FLEXIBLE JOB SHOP PROBLEM

CH. RATNAM[1], M. B. S. SREEKARA REDDY[2], PESARU VIGNESHWAR[3], E. V. RATAN DEEP[4] and K. NAVEEN KUMAR[5]

[1]Andhra University
Visakhapatnam, Andhra Pradesh, India

[2,3,4,5]K L University
Guntur, Andhra Pradesh, India

Abstract

In this paper, a new hybrid evolutionary algorithm is proposed to solve a multi objective flexible job shop scheduling problem (MOFJSSP). Here hybridization is done between the Non-dominated Sorted Genetic Algorithm (NSGA-II) and the Simulated Annealing (SA) algorithm. This paper also proposed a new priority based algorithm, according to this it is implemented in the initialization of chromosomes from the randomly generated initial population. This step is meant for removing insignificant initial solutions to improve the performance of the proposed hybrid algorithm. Here the minimization of makespan time, total workload, and critical workload are considered as performance measures. The proposed hybrid algorithm with priority based initialization is evaluated by using kacem instances and a comparison with other existing approaches shows the effectiveness of proposed algorithm.

1. Introduction

In the field of operations research production planning scheduling and optimization are important problems in which it deals with assignment, and routing sequences for operations to be done to meet the demands with in stipulated time. As it is well known that classical job shop scheduling is one of the most important issue because of its complexity and it is considered to be stubborn classical problem since 1960 and hence it is known to be very

2010 Mathematics Subject Classification: 68N13, 97N50.
Keywords: Hybrid Algorithm, Multi Objective, Simulate Annealing, Non-dominated sorted Genetic.
Received March 10, 2017; Accepted July 20, 2017

high NP-hard Problem to solve as far as it concern [1]. Extension to this a new era has been started known as Flexible job-shop scheduling system which has enhanced the required demand with in time constraints thereby satisfying overgrown customers [2]. But still it is also considered more complex problem than JSP due its major involvement tasks i.e. assigning operations to machines besides sequencing the operations of machines on to machines.

Dispatching rules were first implemented to generate initial population by taking makespan time minimization has a primary objective by Brandimarte [3]. He also introduced routing decision fixed in the initial solution to obtain the job shop scheduling problem. After a couple of years Paulli, solved this combinatorial problem through hierarchical approach [9]. Xia and Wu implemented hierarchical method by using swarm intelligence to assign operations over to machines and simulated annealing has sequential operation on machines [4]. Hurink et al. introduced the new beam search algorithm to generate the initial Solution. He also reduced the neighborhood size and changed the routing and sequences of operation on machines through Tabu search, which helped in finding the best quality solution in least amount of time [5].

Vahit et al. proposed an object-oriented approach for multi-objective job shop scheduling problem and proved that by using objective oriented programming not only enhanced the effective control system but also achieving effective solutions [6]. Brucker and Schlie developed a polynomial algorithm for solving the FJSP with two jobs [7]. Pezella et al. proposed a genetic algorithm, which was implemented to exploit a lot of domain knowledge to generate the initial population, this was done through localization approach [8]. A multistage based genetic algorithm was proposed by Zhang and Gen to solve the FJSP [10] and later Saidi Mehrabad and Fattahi implemented tabu search along with mathematical modelling with dependent sequences in flexible job shop scheduling problem [11].

A single objective and multi-objective case FJSP was studied through the proposed genetic algorithm controlled by the assignment model which was generated by the localization. They used this integrated approach to solve the problem [12-13]. A multi-start procedure and tabu search which is considered as a hierarchical procedure was used to solve the FJSP by Rigao [14]. Xia and

Wu used the Particle swarm intelligence and simulated annealing based hybrid algorithm [15]. A multi-objective decision making and the global criterion methods were used to solve the FMS scheduling problem with considering the three objectives [16]. Based on the objectives and their relations Zhang Zheng and Wu solved the MOFJSP through hybridization algorithm between the Ant colony optimization and Particle swarm intelligence [17]. A bottle neck shifting i.e. with the innovative local search algorithm along with genetic algorithm was implemented by the Gao, Gen, Sun and Zhao [18].

However, due to its complex nature involvement it has been approximately solved through plethora of metaheuristic algorithms such genetic algorithm, tabu search, swarm intelligence techniques such as Ant colony Optimisation, particle swarm algorithm where these tools were directly implemented without any real world scenarios [14]. In order to satisfy multi-objective criteria, in this paper we would like to emphasize more on two main objectives, this particular point where in our research a primary weightage is given i.e. by keeping a view on real world scenarios.

Primarily our research work is focused on the multi-objective criteria, as from the past many years a plethora of research activities were being done in solving single objective function in the FJSP, commonly the makespan as main function. But in real world the FJSP need to satisfy multiple criteria scenarios i.e. due date related operations need to perform apparently more relevant for the decision makers in order to maintain name and fame. Therefore to maintain in-time delivery minimization of tardiness with multiple criteria are required according. As long as scheduling is concerned tardiness is said to be delay in the actual completion time. If the job is completed well before the due date or on due date then it refers to zero tardiness. But in multiple criteria in most of the cases zero tardiness is highly impossible by satisfying all these multi-objectives. In this paper we have attempted to resolve this problem by considering multi-objectives.

Second, as long as most of the research works are concern till this period most of them emphasized by taking ideal cases into consideration, by solving this combinatorial problem through their developed algorithm or through the classical algorithms, but hybridization algorithm which maintains harmony between the exploration and exploitation is concerned the available literature review was very low in number, in this paper an attempt to solve this multi-objective combinatorial problem through the hybridization between the two well-known algorithms i.e. Non-dominated Genetic Algorithm and Simulated Annealing is implemented to maintain harmony between the well equilibrated exploration and exploitation assuming there will be zero disruptions. However there are many cases on the shop floor where there are unavailability of machines is a common event due to the plethora of operations on that particular machine which may results into machine break downs, and the operations which are scheduled on that machine need to be resumed until the machine is get repaired or replaced, it not only effect that particular job operation in FJSS but in fact it will show its effect on all job operation resulting into maximum tardiness. At present this paper attempts to solve only the multi-objective criteria model without concerning about the above issues, in future aspects the above problems will also be considered.

2. Formulation of the MO-FJSP

The MO-FJSP can be formulated as the following.

1. Consider m machines $M = \{M_1, M_2, M_3, \ldots, M_m\}$ and n independent job $J = \{J_1, J_2, J_3, \ldots, J_n\}$.

2. Each Job has n_j operations formed by a sequence of $\{o_{i1}, o_{i2}, \ldots, o_{i,nj}\}$.

3. Each operation of job J_i should be processed through one machine from the set available machines for that particular job.

4. The processing time of the operation is machine dependent. And $D_{i,j,k}$ be denoted as processing time of $o_{i,j}$ on machine M_k.

5. Denote the completion time of operation $o_{i,j}$ by $c_{i,j}$. The preceding constraints is given as $c_{i,j} > c_{i,j-1}$.

6. All the jobs are ready at time zero, in other words $c_{i,j} = 0$.

The scheduling consists of two sub problems: the routing sub problem that assigns each operation.

To an appropriate machine and the sequencing sub problem that determines a sequence of operations on all the machines. Let be the completion time of job is the summation of processing time of operations that are processed on machine. Three objectives, namely, makespan, total workload, and Critical workload are to be minimized in this paper, which are defined respectively as follows:

$$\textit{makespan}\ (M,\ S) : f_1 = \max_{j=1,\ldots,n} C_{jn_j} \tag{1}$$

$$\textit{total workload}\ (W_T) : f_2 = \sum_{k=1\ldots m} \sum_{o_{ji} \in o_k} D_{jik} \tag{2}$$

$$\textit{critical workload}\ (W_c) : f_3 = \max_{k=1,\ldots,m} \sum_{o_{ji} \in o_k} D_{jik}. \tag{3}$$

Moreover, the following assumptions are made in this study: all the machines are available at time 0; all the jobs are release at that time 0; each machine can process only one operation at a time; each operation must be completed without interruption once it starts; the order of operations for each job is predefined and cannot be modified; the setting up time of machines and transfer time of operations are negligible. For illustrating explicitly, a sample instance of FJSP is shown in Figure 1, where rows correspond to operations and columns correspond to machines.

3. Multi-Objective Hybrid-Evolutionary Algorithm for FJSS

3.1. Brief Description of NSGA-II

Since there are three objectives included in equations (1), (2) and (3) and flexible job-shop scheduling problem is very well known as NP-hard problem, the application of multiple objectives heuristics algorithm are very much essential in solving. It is well known fact that Non-dominated sorting genetic algorithm-II (NSGA-II) is very efficient from many previous studies. The main procedure of NSGA-II can be briefly described next.

It is known that without loss of generality, the i^{th} generation of NSGA-II is considered. It basically works on population generation known as initialization of population P_t, later this population acts as parents to the new class of child Q_t, which are generated through genetic operations. The newly formed child population combined with the parent population $R_t = P_t U Q_t$ with size 2N. With the non-dominated sorting criteria a new set of population is generated of size N, by taking the number of objectives into consideration such $F_1, F_2, \ldots, F_n$. The two main criteria which made the NSGA-II special in its various applications is that its crowding distance criteria and sorting ranks the solutions in different pareto fronts.

3.2. Brief Description of SA

Simulated annealing algorithm has produced its innovative results in various applications in industry since from its introduction through efforts of Kirkpatrick Gelatt. It has been started applying to job-shop scheduling from 2005 and it still continuous in its innovation (Xia and Wu 2005). SA algorithm start with initial solution or a chromosome and it computes a new solution through probability criteria which is given as:

$$y = \min(1, \exp\{-f(x^1, x)/T\}), \tag{4}$$

where $f(x_i, x) = f(x_i) - f(x) \cdot f(x)$ is the objective function with the chromosome x as a input to the function, in this case our objective function is the makespan time, total work load and critical workload. For which the loop is made to run three times for every temperature value $f(x) \leq f(x^1)$. If the new solution x^i is better than the previous solution x, then the new solution x^i is accepted with probability of x, the higher the temperature T, the better for the optimal solution is being searched in the current point, as the temperature decreases, it approaches to the optimal solution. It is also well known fact that simulated annealing process is a iterative process, with learning rate (α) as one of very effective parameter in controlling the local optimal, in the case of non-convex function, and it also avoids the boundedness of its temporary solution.

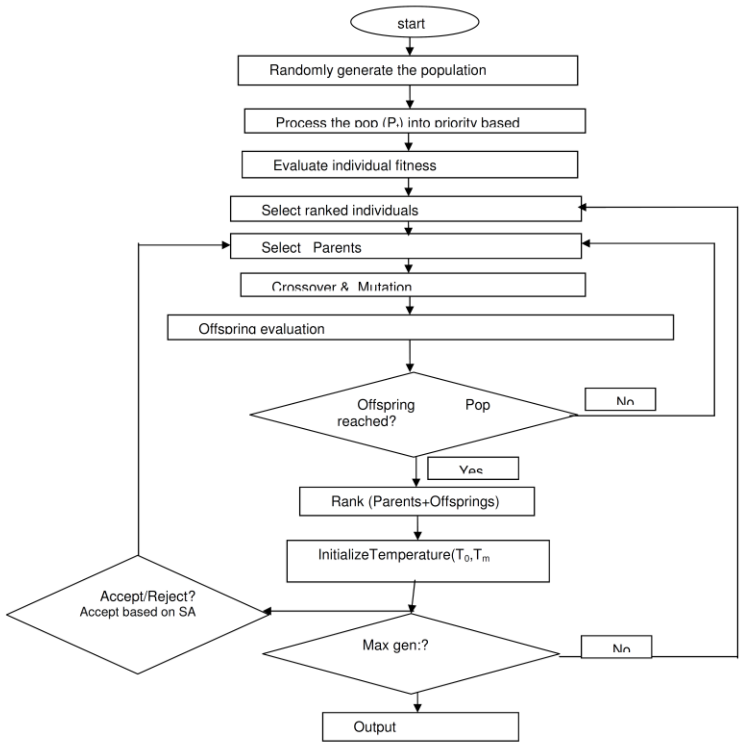

Figure 1. Flow-chart of Proposed Methodology adopted.

4. Multi Objective Flexible Job Shop in Detail

4.1. A new Hybrid Evolution Algorithm

The frame work for the proposed new hybrid evolution algorithm was formulated based on the originality of NSGA-II (1) and uniqueness of simulated annealing (2). In this framework an initial population with N chromosomes have been created randomly, and further these randomly generated chromosomes where processed into priority based algorithm, according to this the operations of particular job have been adjusted based on its priorities. The following is the chromosome before priority based operation and after the operation.

O	O	O	O	O	O	O
3-1	2-2	3-2	2-1	2-3	1-2	2-2

An instance of randomly generated chromosome The newly created chromosomes which contains a sequence of genes, represents the job, operation, machine respectively will be processed by machine K which are assigned initially based on its allotment are further processed into evaluation of objective functions i.e., makespan time, total work load, and critical work load. Now each such chromosomes are processed into the new hybrid evolution algorithm (NSGA-II+SA). The following are the steps to implement this algorithm.

O	O	O	O	O	O	O
3-1	1-1	3-2	2-1	2-2	1-2	2-3

An instance of generated chromosome after priority based algorithm

4.2. Chromosome Encoding

The initial solution of the flexible job shop can be encoded through assignment of operations to the machines and processing the priority based operations on to the machines. Therefore it consists of two vectors, they are machine assignment vector and sequence of operations on to the machines vector as indicated in above figure where it depicts both the vectors. Taking the figure as an example, it encodes the sequence of seven operations, two of job 1, three of job 2, and two of job 3. The first operation of job 3 was processed on machine 3 and the first operation of job 2 was processed on machine 2 and first operation of job 1 was processed on machine 1, which has lowest priority. Therefore the first operation of job 3 has highest priority.

4.3. Chromosome Decoding

Chromosome decoding involves the allotment of period of time on its assigned machines for each operation according to the chromosome generated one by one in their order. The selection of machine based on its availability was done dynamically. When one operation was treated from its selected machines then the idle time intervals between the operations that already been schedule on that particular machines was scanned from left node to right node along with the maximum completion time of the previous operation of that particular job was also considered. The following is the mathematical representation of chromosome decoding.

$$\left.\begin{array}{r} \max\{A_x, t_{i,j-1}\} + q_{i,j,k} \le F_x \\ A_x + q_{i,j,k} \le F_x \end{array}\right\} \begin{array}{l} \text{if } j \ge 2 \\ \text{if } j = 1 \end{array}$$

Let $a_{i,j}$ be the starting time of the operation $P_{i,j}$ in a schedule and $t_{i,j}$ its completion time has an operation can be initialized only after the completion of its precedent operation of that job. When $P_{i,j}$ to be allocated within the availability of the time interval $[A_x, F_x]$, $\max\{A_x, t_{i,j-1}\}$ if $j \ge 2$ or A_x if $j = 1$ is taken as initial time. It will be arranged at the end of M_k if there is zero existence of the interval. The schedule which is generated by this type of decoding technique is known to be an active schedule.

4.4. Initialization

The initial population is generated of size N_{pop} known to be initial solutions were decoded. The so form decoded solutions are evaluated through the desired objective functions and set these generation number as $g = 1$. Though there are different kinds of population generation techniques for the successfully implementation of evolutionary algorithm a good initial population is the one which locates best promising areas and provides optimal diversity by preventing the local convergent optimal solution. The AL proposed by Kacem et al. (2002a) has been a common method for making the routing decision for the initial population (Pezzella, et al., 2008; Yazdani et al., 2010; Zhang et al., 2011) as per the operation sequencing, Pezzella et al., 2008 used the three dispatching rules.

In our algorithm we proposed a priority based approach for the initial population generation after the random chromosomes according to this algorithm a diversified machine assignment vector is implemented this can be very well understood through the below depicted pseudo code. According to this group of sets were created depending on the number of jobs in the sample case which is presented as case study-1. These 3 sets were created as S1, S2, S3 consists of operations of their particular jobs. The random chromosomes which are initialized undergoes the following codes to form new population by checking its preceding operations and succeeding operations.

4.5. Reproduction

The N_{pop} off springs were generated by mating selection and crossover. Of these newly generated off springs and the original off springs with their sum equals to N_{pop} individuals survive to generation equals to $g+=1$ by natures selection. Basically, obtaining the best optimal results in NSGA-II depends on the criteria such as population size, selection type, crossover fraction and mutation functions along with crowding distance and ranking parameter of the non dominating solution set which is further used in SA. The chromosomes considered in this work includes, population size 50, 100 and 150, tournament selection, crossover probability is 0.8 and mutation. function is adaptive feasible. The parameter setting for these criteria was made by the process of trial and error method to obtain the optimal result that is expected from this study an equilibrated exploration and exploitation.

The diversity of population to avoid local convergence of the optimal solution is maintained through effective hybridization between NSGA-II and simulated annealing is carried. It is observed that lots of diversity was occurred by the repetitive occurrence of similar individuals. In this paper we proposed hybrid algorithm to make use of the information provided by the similar individuals on one hand and exploiting these individuals through simulated annealing based on mark overs principle and exploration is done through NSGA-II algorithm on other hand.

4.6. Termination

This is the final step implementation of algorithm if $g > G$ otherwise $g+=1,$ go to step 4 i.e., reproduction.

4.7. Pseudo code for Priority based algorithm

```
for i- > 1 : n
{
    for j- > 1 : m
        {
        if x(i) equals s1(j)
```

$p\,o\,s = j,\ key = i$

while $p\,o\,s > 0$

{

$pos--$

$x(i) = s1(pos)$

}

$j++$

}

while $(x(i--)$ not equals zero is equals $(x(i--), x(key))$ while (1)

{

$x(i) = s1(pos++)$

}

$i--$

}.

5. Case Study I

We tested the priority based method of population generation and multi objective hybrid evolutionary algorithm i.e. with NSGA-II and SA for the innovative exploration and exploitation of global optimal solution with in the intelligent selection through various case studies of these we are presenting a basic case study in detail to a few Kacem et al. sets [12-13].

Consider the following flexible job shop schedule. There are three jobs of seven operations which are to be processed on three machines. Each operation of a particular job were processed on to their assigned machines which is done dynamically. The processing time of each operation of the jobs is presented in table Table No-1 which is pre determined. This pre determined operation chart was employed in to the matlab in the form of .XLS format. This excel sheet is used to read the operation time of that particular operation. The parameters of the algorithm are as follows: a population size of 100, a maximum generation of 50, crossover probability of 0.7 and mutation probability of 0.05 is maintained in NSGA-II case while the number of iterations were maintained as 100 and temperature maintained was 50 in the case of simulated annealing. The best solutions which are obtained based on ranking parameter of NSGA-II where further processed in to simulated annealing for exploration. This hybrid algorithm was run 10 times to obtain total makespan, total workload, and critical workload.

The calculated results and the solution distributions are presented in Table 1 the optimal Pareto front is constituted by the points set it includes $O_{3-1}O_{2-1}O_{1-1}O_{2-2}O_{1-2}O_{2-3}O_{3-2}$. It obviously reflects from the solution set that there is a significant trade-off between the makespan time, total work load and critical work load.

A table job shop problem is considered with three jobs, seven operations and three machine, which is depicted below.

Job	Operation	M1	M2	M3
J1	O_{1-1}	2	0	4
	O_{1-2}	5	3	3
J2	O_{2-1}	0	3	5
	O_{2-2}	2	3	4
	O_{2-3}	0	0	3
J3	O_{3-1}	5	3	4
	O_{3-2}	2	0	5

Figure 1. 3X3 instance.

The Pareto optimal solutions of the considered two performance measures with the classical hybrid algorithms and proposed hybrid NSGA-II algorithm is shown. Table 3 depicts different instances is shown in the below sections, before that the Table depicts the minimum makespan, total workload, critical workload for the corresponding chromosomes and their comparison results. For 3 jobs 3 machines instance with hybrid NSGA-II algorithm the optimum makespan value is 8, similarly the critical workload is calculated to be 7 and the total workload is calculated to be 19. Though there are different chromosomes satisfying these values, hence clearly the flexible job shop definition can be depicted. This trend is continuing and the results are improving for proposed approach. All the mentioned algorithms are coded with MATLAB software and the problem is tested on Intel® Core™2 Duo CPU T7250 @2.00GHz, 1.99 GB RAM.

Table 1. Solution Table for 3X3 Instance.

Chromosome	Makespan	Critical Work Load	Total Work Load
O_{3-1} O_{1-1} O_{3-2} O_{2-1} O_{2-2} O_{1-2} O_{2-3}	9	9	21
O_{3-1} O_{1-1} O_{3-2} O_{2-1} O_{2-2} O_{2-1} O_{1-2}	10	8	20
O_{3-1} O_{1-1} O_{2-1} O_{3-2} O_{2-2} O_{1-2} O_{2-3}	9	9	21
O_{1-1} O_{3-1} O_{2-1} O_{3-2} O_{2-2} O_{1-2} O_{2-3}	9	9	20
O_{2-1} O_{1-1} O_{3-1} O_{2-2} O_{3-2} O_{2-3} O_{1-2}	8	7	19
O_{2-1} O_{1-1} O_{3-1} O_{2-2} O_{3-2} O_{1-2} O_{2-3}	8	7	19
O_{2-1} O_{1-1} O_{2-2} O_{3-1} O_{2-3} O_{1-2} O_{3-2}	8	7	19
O_{3-1} O_{2-1} O_{1-1} O_{2-2} O_{1-2} O_{2-3} O_{3-2}	8	7	19
O_{3-1} O_{3-2} O_{2-1} O_{1-1} O_{2-2} O_{1-2} O_{2-3}	9	9	21
O_{1-1} O_{1-2} O_{3-1} O_{2-1} O_{2-2} O_{3-2} O_{2-3}	11	9	24
O_{1-1} O_{2-1} O_{2-2} O_{3-1} O_{1-2} O_{2-3} O_{3-2}	8	8	20

6. Case Study II

In this case study we consider four sets of benchmark data from Kacem et al. [12-13] which includes 8X8, 10X10, 15X10. The parameters of this case study follows as usual from the above case study i.e., the population size of 100, maximum generation of 50, the cross over probability of 0.7, mutation probability of 0.05. The hybrid algorithm was run for 10 times in each case to obtain the all the mention objectives. The results of these benchmark problems are shown in table where *M* represents minimum Makespan time, T.W represents minimum total work load, C.W represents minimum critical work load.

6.1 Solution Table: A. The relative comparison is being made with the other existing hybrid algorithm, of Kacem et al. sets from the literature preview. Clearly from the table for 8 machines and the 8 jobs instance, the new hybrid algorithm could estimate the Pareto optimal solution as 14 as Makespan time 11 as critical workload (Wc) and 76 as Total workload (T.W.L), where as in the case of AL+CGA algorithm the optimal solution obtained to be the 14, 12, 77 respectively where our new algorithm is 1 unit less than the critical workload and total workload. The below presented graph clearly shows the improvement in the Makespan (M.S), Total workload, and Critical work Load.

Table 2. 8X8 Instance solution table.

AL+CGA(1)	PSO+SA(2)	PSO+TS(3)	NSGA-II+SA(4)	Objectives
15	14	15	14	Makespan
13	12	12	11	Max(Wc)
79	77	75	76	T.W.L

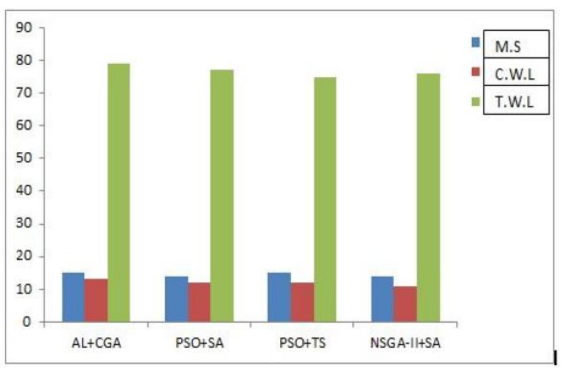

Figure 2. Bar Graph for 8X8 Instance.

M.S=Makespan; C.W.L=Critical work load; T.W.L=Total work load

The PSO+SA algorithm could able to estimate the best optimal solution to be 15, 12, and 75 respectively which is also higher than the proposed algorithm. The PSO+TS algorithm optimal solution were also more when compared with the new proposed algorithm. Though there may be single objective case might be the best but when the over-all multi-objective is concerned this newly proposed algorithm is considered to be the best.

B. From the table for 10 jobs and 10 machine instance, the new hybrid algorithm could estimate the Pareto optimal solution as 7 as Makespan time 5 as critical workload (Wc) and 43 as Total workload (T.W.L), where as in the case of AL+CGA algorithm the optimal solution obtained to be the 7,5,45 respectively where our new algorithm is lesser than the critical workload and total workload.

Table 3. 10X10 Instance solution table.

AL+CGA	PSO+SA	PSO+TS	NSGAII+SA	Objectives
7	7	7	7	Make Span
5	6	6	5	Max(Wc)
45	44	43	43	T.W.L

Table 4. 15X10 Instance solution table.

AL+CGA	PSO+SA	PSO+TS	NSGAII+SA	Objectives
24	12	11	11	Makespan
11	11	11	10	Max(Wc)
91	91	93	91	T.W.L

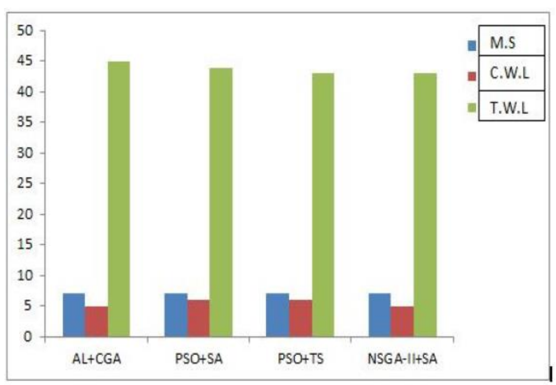

Figure 3. Bar Graph for 10X10 Instance.

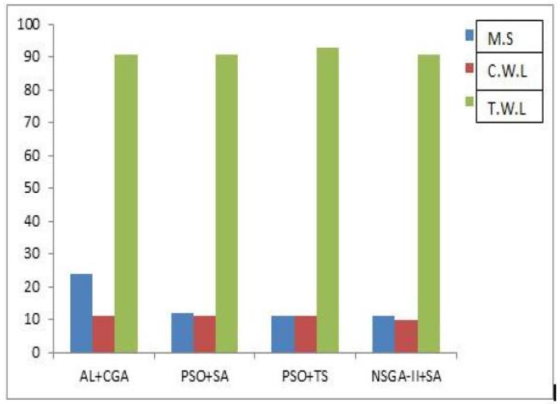

Figure 4. Bar Graph for 15X10 Instance.

M. S=Make span; C.W.L=Critical work load;

T.W.L=Total work load.

The PSO+SA algorithm could able to estimate the best optimal solution to be 7,6,44 respectively which is also higher than the proposed algorithm. The PSO+TS algorithm optimal solution were also more when compared with the new proposed algorithm i.e. 7, 6, 43. Though there may be single objective case might be the best but when the over-all multi-objective is concerned this newly proposed algorithm is considered to be the best. C. Clearly from the table for 15 jobs and 15 machine instance, the new hybrid algorithm could estimate the Pareto optimal solution as 11 as Makespan time 10 as critical workload (Wc) and 91as Total workload (T.W.L), where as in the case of AL+CGA algorithm the optimal solution to be the 24,11,91 respectively where our new algorithm is lesser than the critical workload and total workload. The PSO+SA algorithm could able to estimate the best optimal solution to be 12, 11, and 91 respectively which is also higher than the proposed algorithm. The PSO+TS algorithm optimal solution were also more when compared with the new proposed algorithm i.e 11, 11, 93. Though there may be single objective case might be the best but when the over-all multi-objective is concerned this newly proposed algorithm is considered to be the best.

7. Results and Discussions

This paper has studied the multi objective flexible job shop scheduling problem by considering all the major objective cases, namely makespan time, total work load, critical work load. Which are the major scenarios occurring in any real world manufacturing systems, a plentiful of research was being done to address this problem with the new innovative ideas in the evolutionary algorithms or through swarm intelligence. Though there are too many solutions approaches to these kinds of the problem every researcher had his/her own implementation of their innovations to improve the process. It is well known fact that an optimal solutions can never be obtained in fact a closeness to optimal can be obtained through heuristics. To propose the best optimal approach to this problem, we have developed a new Hybrid evolutionary algorithm through NSGA-II and simulated annealing based on their uniqueness in their functioning. We first conform the generation of initial solution through innovative technique named Priority based according to it the randomly generated populations are arranged into a significant populations thereby decreasing the insignificant solutions, enhancing the computation times based on the input data, where it has maintained operation sequences in well diversified manner by scanning the precedent and succeeding operations of that particular job.

In this hybrid algorithm we first adapt the NSGA-II and SA to flexible MO-FJSP prior to the interface of both the algorithms. Later the flexible MO-FJSP was carried out using the hybrid NSGA-II and SA through well defined chromosome encoding, decoding, and initialization techniques. It is to be well noted that the initialization technique carried out was innovative, which has uncommon literature available in the operations society.

The novelty of this paper lies in the well equilibrated exploration and exploitation which is carried out based on the intelligent interface technique between both the evolutionary algorithms. In this strategy the exploration was carried by the NSGA-II, with its unique strategies of ranking parameter and crowding distance, according to it the best optimal solution which are obtained on the top most set were considered for the further process into the SA case. In SA case, the exploitation was carried out through acceptability and rejection criteria principle based on this principle the best optimal solution was obtained. Our hybrid evolutionary interface technique is the only one which could maintain the well diversity in its population generation, exploration, and exploitation when compared to the first adapted technique of pure NSGA-II, and SA to the MO-FJSP with well performed bench mark sets.

For the future work, this so formed developed algorithm need to undergo plentiful of test cases to improve its efficiency, thereby decreasing the CPU time, and space complexity.

References

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

[2] Y. Mati and X. Xie, The complexity of two-job shop problems with multi-purpose unrelated machines, European Journal of Operational Research 153 (2004), 159-169.

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

[4] W. Xia and Z. Wu, An effective hybrid optimization approach for multi-objective flexible jobshop scheduling problems, Computers Industrial Engineering 48(2) (2005), 409-425.

[5] J. Hurink, B. Jurisch and M. Thole, Tabu search for the job-shop scheduling problem with multi-purpose machines, OR Spektrum 15 (1994), 205-215.

[6] Vahit Kaplanog, An object-oriented approach for multi-objective flexible job-shop scheduling problem, Expert Systems with Applications 45 (2016), 71-84.

[7] A. Baykasoglu, L. Ozbakir and A. Sonmez, Using Multi-objective Tabu search and grammerstomomodel and solve multi-objective flexible job shop scheduling problems, Journal of Intelligent Manufacturing 15(6) (2004), 777-785.

[8] F. Pezzella, G. Morganti and G. Ciaschetti, A genetic algorithm for the flexible job-shop scheduling problem, Computers and Operations Research 35 (2008), 3202-3212.

[9] J. Paulli, A Hierarchical Approach for the FMS Scheduling Problem, European Journal of Operational Research 86 (1995), 32-42.

[10] H. Zhang and M. Gen, Multistage-Based genetic algorithm for flexible job shop scheduling problem, Journal of Complexity International 11 (2005), 223-232.

[11] M. Saidi Mehrabad and P. Fattahi, Flexible job shop scheduling with Tabu search algorithm, International Journal of Advanced Manufacturing Technology 32 (2007), 563-570.

[12] I. Kacem, S. Hammadi and P. Borne, Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problem, IEEE Transactions on Systems, Man and Cybernetics, Part C 32(1) (2002), 1-13.

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

[14] C. Rigao, Tardiness minimization in a flexible job shop: a Tabu search approach, Journal of Intelligent Manufacturing 15(1) (2004), 103-115.

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

[16] C. Low, Y. Yip and T. H. Wu, Modeling and heuristics of FMS scheduling with multiple objectives, Computers and Operations Research 33(3) (2006), 674-694.

[17] W. C. Zhang, P. E. Zheng and X. D. Wu, Solution to flexible job shop scheduling problems with capacitated constraints based on ant colony genetic algorithms, Computer Integrated Manufacturing Systems 13(2) (2007), 333-337.

[18] J. Gao, M. Gen, L. Sun and X. Zhao, A hybrid of genetic algorithm and bottleneck shifting for multi-objective flexible job shop scheduling problems, Computers and Industrial Engineering 53 (2007), 149-162.