Flexible Job Shop Scheduling using Hybrid Swarm Intelligence

S. Kavitha, P. Venkumar

Abstract: In the present environment investigation, one of the essential tasks to be solved is scheduling. The most significant issue in the Job Shop scheduling process is the flexibility which is occurred during the manufacturing process. This paper presents the hybridization of swarm intelligence's that is Chicken Swarm Optimization (CSO) and Discrete Fish Swarm Optimization (DFSO) to minimize the makespan, overall workload and utmost workload of the machine. The purpose of individual operators is employed to upgrade the fish position and provoke new fishes that are processing times. The purpose of this technique is to speed up the minimum convergence and trapped in the local optimum. The proposed hybrid algorithm results are compared with conventional and existing optimization approaches for a multi-objective flexible JSP process.

Keywords: Flexible Job shop Scheduling, Hybrid algorithm, Multi objectives, Swarm intelligence and processing Times.

I. INTRODUCTION

Flexible Job Shop Scheduling Problem (FJSSP) is an accumulation of the standard JSP that established a task to be processed every machine from the available machine set. It populates most of the challenges and complexities [1] of its predecessor JSP and is more obscured than JSP due to the expansion needs to decide the task of activities to machines [2]. To tackle the problem, more consideration was applying the intelligent algorithm in the late decades. The behavior of swarm intelligence analyzed by a few researchers and depicts some strategies for separating job shop scheduling [3]. In the previous two decades, nature – motivated algorithms keep improvements in the problems of solving optimization [4]. There are extended to comprehend an extensive variety of utilization, for example, scheduling optimization problems [5]. The probabilistic search techniques observed by most analyst and they proposed that it is an attiring another option to find solution this constrained optimization problem such as Genetic based algorithms [6], Simulated annealing algorithm, Bat algorithm [7] particle swarm optimization [8], Ants colony algorithm [9], Artificial immune algorithm, etc. An altered coding scheme was introduced to minimize the makespan measure of FJSSP [10]. In the initial phase, a novel machine task methodology was furnished and while the current solution had not been enhanced at the time an enhancement procedure was performed [11]. The main aim of this study is the scheduling steadiness in the FJSSP in that machine breakdowns are normal. The simulations of a machine breakdown can provide results in a scheduling solution with higher security, more robustness and nearest to the real world and it has not been researched in many investigations [12]. An optimal schedule neglected the assignments of tasks machines and completion time in JSSP. In this research, a hybrid algorithm is optimized for an FJSSP because it apposing multi goals like maximum workload, total workload and Markesan time JSSP is tackled by the behavior of hybrid optimization as CSO and DFSO are connected.

II. LITERATURE REVIEW

Literature Details	Technique	Objectives	Description
Chao Peng et al. 2018 [13]	GA	Make span Time	JSSP in realistic production exercises dependably needs to think about the requirements of various manufacturing assets. Both machines and workers are considered during the process of job shop scheduling.
TIANHUA JIANG et al.2018 [14]	GWO	Completion Time	They developed a GWO algorithm with an objective of minimizing the maximum completion time.
Liang Xu et al. 2017 [15]	Hybrid cloud particle swarm optimization (HCPSO)	Make span, Total Work Load and Maximum Workload	It is gone for reducing the completion time of jobs, total workload, and maximum workload. The weight has stable propensity and arbitrariness properties in view of the cloud model, which enhances the convergence speed, as well as maintains the variety of the populace.
Song Huang et al. 2018 [16]	Particle Swarm Optimization (PSO)	Make span, Total Work Load and Maximum Workload	The PSO is used to solve multi-objective optimization problems.

Revised Manuscript Received on December 15, 2019.

* Correspondence Author

S. Kavitha, Department of Mechanical Engineerig, School of Automotive and Mechanical Engineering ,Kalasalingam Academy of Research and Education Krsihnankovil, Tamilnadu, India, kavitha@klu.ac.in

***P.Venkumar,** Department of Mechanical Engineerig, School of Automotive and Mechanical Engineering, Kalasalingam Academy of Research and Education, Krsihnankovil, Tamilnadu, India, p.venkumar@klu.ac.in

Kamatchi et al.2017 [17]	Hybrid optimization (Firefly and Cuckoo Search)	Make span	The tensile, open shop scheduling is known to be NP-hard. Cuckoo Algorithm (CA) is one of the generally utilized strategies for constrained optimization. What's more, it gave the best outcomes in contrast with different algorithms. A drawback of CA is that they effortlessly become trapped in the neighborhood minima
Jun-Qing Li et al. 2013 [18]	Discrete artificial bee colony algorithm (DABC)	Make span, Total Work Load and Maximum Workload	The DABC algorithm is used to solve the scheduling problem
Thi-Kien Dao et al. 2015 [19]	Bat algorithm	Make span	In the analysis, 43 scheduling dataset problems were tested using the proposed bat algorithm.

In FJSSP and JSSP the remaining manuscript is structured as takes after section. Section 2 presented our approach and section 3 analyzed the implementation and comparison results. Finally, the proposed work conclusion part is given in section 4.

III. HYBRID ALGORITHM FOR FJSSP: A METHODOLOGY

The knowledge of different swarm optimization model is considered for the FJSSP process. In FJSSP, every job is executed on the machines with machining time in a specific order. In this model, each job is processed only one machine at a time. Ordinarily, for the scheduling purpose, there are M machines and N jobs utilized. The considered jobs are processed with various routes or sequences. Hence, the difficulty of scheduling jobs is based on the number of machines, jobs, and sequences of jobs. Our proposed methodology considers Multi-Objective functions in the number of jobs and machines (J*M) process, for analyzing purpose AFSO with CSO used. From this implementation and mathematical procedure to solve our objectives in FJSSP, the detailed discussion of our methodology is described in beneath sections.

A. Problem Definition

JSP is one of the combinatorial optimization problems. The sequence of operations performed in existing engines determined the scheduling problem with a minimized processing time that needs for the process completion [20]. The number of machines symbolized as 'm' and the different number of jobs symbolized as 'j', and they are produced to be scheduled. At a particular given time, each job contains a number of the process which should be executed on a particular machine. In Table 1 furnished the sample processing time.

Table-I: Processing Time of Benchmark problems

Jobs/Machines		M1		M2		M3		M4
J1	O11	3	O12	4	O13	5	O41	6
J2	O21	2	O22	8	O23	4	O42	3
J3	O31	1	O32	9	O33	10	O43	6
J4	O34	10	O34	6	O34	5	O44	7

Also, the sample processing time appears in table 1. At the point, the operation will be handled in a particular machine, it is feasible for the operations to queue up ahead of time so that the machine is being utilized by another operation.

B. Function for FJSSP Objective

Proposed FJSSP process considered three objectives such as Make span time (F1), total workload (F2), and Machine Work Load (F3). For the investigation, a set of machines and jobs, it's denoted by $m_i = \{m_1, m_2, m_3, ...m_n\}$ and $j_i = \{j_1, j_2, j_3, ...j_n\}$ respectively. In FJSSP, each job j consists of a sequence of nj operations that is $\{O_{1j}, O_{2j}, O_{3j}, ...O_{nj}\}$ each operation has to be executed to finish a job. All objective functions constrained as a minimum its criteria followed as:

F1: Make span Time $\Rightarrow Min\, F1 = \underset{1 \le i,k \le n}{Max} \{T_{j*m\_time}\}$ (1)

F2: Total Workload $\Rightarrow F2 = \sum_{ink=1}^{m,n,n_i} T_{ink} * P_{ijk}$ (2)

F3: Maximum work load $\Rightarrow F2 = \max_{1 \le k \le m} \sum_{in=1}^{m,n,} T_{ink} * P_{ink}$ (3)

The objective function (F_{obj}) i.e. fitness function of the proposed model can be written as

$$F_{obj} = MIN(F1, F2, F3) \quad (4)$$

Statement and Description of Our proposed FJSSP as

For easy analysis, in the assignment task, sub-problem assigns each operation to an appropriate machine.

One of the sequencing sub-problems calculates an order of operations on each machine.

Furthermore, the embraced theories in this problem are:

If all the machines are 'm', it accessible at time zero, then all jobs are N, it also organized for processing at time zero.

There are no priority limitations in the operations of various jobs.

The operation processing time of each machine is characterized ahead of time.

During the processing time, the interruption of each machine is negligible.

Parameter Notation

m or M Number of machines

n or N Number of Jobs

k n Number of the machine in $k\,th$ the job

O_{ij} Operation of n machine with n jobs

P_{ink} Processing Time of each job with each machine

T_{j*m} Job completion Time for each O_{ij}

C. Hybridization of Swam Optimizations

The presented swarm intelligence optimization is one of the Metaheuristic approaches which works on the principle of swarming behavior of the population. There are cooperation and competition between initialized individuals, which introduced an algorithm, attains the optimal solution among the constrained problem. The presented swarm intelligence optimization approach is an advanced technology which mainly focused by many scholars. And also the number of researchers has proposed many swarm intelligence algorithms like CSO and DFSO. The Hybridized form of CSO and DFSO is shown in figure 3 and is utilized in the proposed work to solve our objective function that is the minimum of J*M performance.

D. Chicken swarm (CS) Optimization

In the CS and the behavior of the chicken swarm, the mimics of CS optimization are in the hierarchal order derived from the perception of the winged animals foraging behavior and it is presented in figure 1.

Rooster: The chickens with the best fitness acted in this group, so it's called as head of the CS model.

Chicks: Chickens with the worst fitness solution act as this group.

Hens: Remaining all solution, expect above groups are called hens or mother.

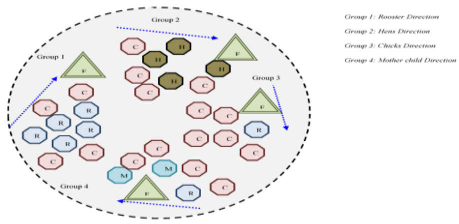


Fig. 1. **CS behavior**

Generally, the relationship between these groups as mother-child with family model randomly established. The food already found by others is stolen randomly by the chickens [20], the chicken with worst fitness value is a search around their mother (a hen) for food, and the dominant individuals have benefit in the struggle for food.

Initial solution generation of j*m model shown in the below equation (5);

$$m_n^k = \begin{bmatrix} m_{11}^k & m_{12}^k & \dots\dots & m_{1n}^k \\ m_{21}^k & m_{22}^k & \dots\dots & m_{2n}^k \\ \dots\dots\dots\dots & \dots. & \dots & \dots.. \\ m_{j1}^k & m_{j2}^k & \dots\dots\dots & m_{jn}^k \end{bmatrix} \quad (5)$$

Where m_{jn}^k represents the priority of operation O_{ij} is the operation of a job j that is an important process on the machine i.

E. Position and Movement updating process

New solution updating process, consider the chickens as the processing time of each job and machine $j_i (1 \le i \le n)$ and $m_k (1 \le k \le m)$. Considering the initial fitness evaluation, group the solution that is the Roosters C_R and worst fitness as C_C roosters with perfect fitting values have priority for food access than the ones with worse fitting values. Position updated formulated as:

F. Exploitation Phase

$$M_{i,j}^{t+1} = M_{i,j}^{t} * (1 + Rand(0, \sigma^2)) \quad (6)$$

$$\sigma^2 = \begin{cases} 1 & \text{if } R_f \le R_f \\ \exp\left(R_f - P_f \big/ |P_f| + \varepsilon\right), \text{other wise} & f \in [1, N], f \ne i \end{cases} \quad (7)$$

Above updating process $Rand(0, \sigma^2)$ is a Gaussian distribution with 0 values and σ as standard deviation and ε as constant. Remaining group i.e. hens (Except the best and worst fitness solution) can follow their roosters for searching food and then randomly stole the food found by others. For solving the above equation (8), (9), formulated below.

G. Global Optimum Solution

$$M_{i,j}^{t+1} = M_{i,j}^{t} + R1 * Rand * (M_{r1,j}^{t} - (M_{i,j}^{t}) + R2 * Rand * (M_{r2,j}^{t} - (M_{i,j}^{t}) \quad (8)$$

$$R1 = \exp\left((R_f - R_{fr1}) / (abs(R_p) + \varepsilon)\right) \quad and \quad R2 = \exp\left((R_{f2} - -R_p)\right) \quad (9)$$

From the above equation, the notation $Rand$ is indicated as a uniform random number also an index of the rooster, which is the ith hen's group-mate, while $fr2 \in [1..N]$ is an index of the chicken (rooster or hen), which is casually selected from the swarm $r1 \ne r2$. With respect to the chicks, they always follow their mother to search for food. At final; model of minimizing $F1, F2$ *and* $F3$ showed in the below equation.

H. Final Local Minimum

$$M_{i,j}^{t+1} = M_{i,j}^{t} + L * (M_{m,j}^{t} - M_{i,j}^{t}) \quad (10)$$

From the above equation, the notation $M_{i,j}^{t}$ stands for the position of the ith chick's mother $L \in (0,2)$ is a parameter, which deliberates that the chick would follow its hen hunting their food. So the solution is obtained based on the encoding scheme.

I. Discrete Fish Swarm Optimization (DFSO)

Generally, a fish can modify its situation constantly with due respect to the environmental condition and the own state position. The movement of individual fish appeared randomly, yet as a sealum, it is a greatly coordinated progress towards the objective and this DFSO flowchart appeared in fig 2. Fishes stay close to the swarm in order to guard or shield fishes from the predators which means it maintains since the distance from neighboring fishes to evade collisions and endlessly search for food [21].

J. Discrete in FSO (DFSO)

In the new discrete FSO algorithm for FJSSP, a solution refers to all the cities of tour consisting which are arrayed differently and the fitness value of the solution refers to the length of the tour. The procedure of this DFSO described in the below section and this innovative model consider four important phases such as

Discrete Solution Initialization

- Fish Prey behavior
- Fish Swarm Behavior
- Fish Follow Behavior

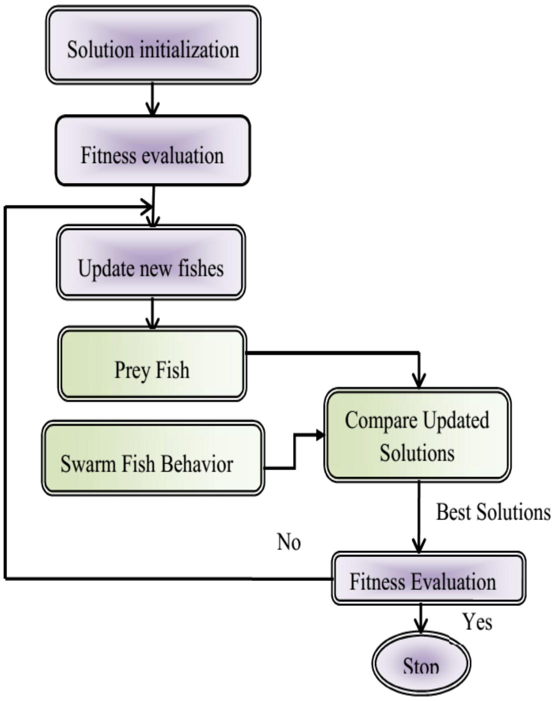


Fig. 2. Flowchart for DFSO

(i) Solution Initialization

FJSSP processing time initialized by the size of benchmark problems and integer values based, it's represented by equation (11).

(ii) Fish Prey Behavior

A group of fish fined food in the process of movement, the other neighbors and deploys to follow and reach the point of these possible. The fish condition analyzed in the ideal state M maxi from visual neighbors, the quantity of partner of M max is if (Dirj < visual) displays that close separation has more nourishment and not very crowded additionally move to the front of M max position; the utilization condition conducted for performing researching generally. The sustenance fixation in this position of fish is expressed as, target capacity esteem. The separation among the artificial fish is $d_{i,j} = ||M_i - M_j||$ here i and j is random fish. Where creates random numbers between 0 and 1 and Step implies the greatest stride size of manufactured fish. Visual is the separate aircraft, the counterfeit fish happens just in the inward range of the hover to the length of the field of vision distinctive acts shown in equation (12) (13).

(iii) Fish Swarm Behavior

Naturally, the assembling in groups leads to guarantee the existence of the colony and to avoid dangers. Likewise, in the fish swarm behavior fishes are assembled in groups in the moving process. Let M_i be the AF current state, M_c be the center position and N_c be the number of its companions in the current neighborhood fish is M_i(Di,j<Visual)a number of counterfeit fish. Based on the highest and lowest fitness functions (F1, F2 and F3) the swarm behavior updated shown in equation (14).

(iv) Fish Follow Behavior

In the process of a group of fish find food, the other neighbors are a move to follow and reach the point of these possible. The condition of fish is investigated its ideal state M_{max} from Visual neighbors, the quantity of partner of M_{max} if (Di,j<Visual) shows that close separation has more nourishment and it's not crowded additionally move to the front of M_{max} position; generally, perform searching conduct by utilizing condition (13).

(v) Move and Leave Process

Fish swim arbitrarily in water; truth be told; they are looking for food or partners in bigger ranges. The leave procedure as stop someplace in water, each AF's behavior result will be the same, the distinction between objective values (food concentration, FC) end up littler within a few cycles, it may fall into local extreme change the parameters haphazardly to the [21] still states for leaping outflow state. These anomalous operations are said in condition (16). Swarm makes fish bound in local extreme qualities move toward a couple of fish keeping an eye on global value, which brings about AF escaping from the local extreme values.

Mathematical equation and Representation in DFSO

Discrete solution: $M_{in} = \{rand(P_i,1)*(ul-lb)+ll\}$ (11)

Prey:

$$M_j = M_i + visual.rand \quad (12)$$

$$M_i^{(t+1)} = I_i^{(t)} + \frac{M_j - M_i^{(t)}}{||M_j - M_i^{(t)}||}.step.rand \quad (13)$$

Swarm $$M_i^{(t+1)} = M_i^{(t)} + \frac{M_c - M_i^{(t)}}{||M_c - M_i^{(t)}||}.step.rand \quad (14)$$

Follow: Shown in equation (15)

Leave and Move: $M_{some}^{(t+)} = M_{some}^{t} + \alpha * rand$ (15)

Above the mathematical equation P_i as processing time and J*M problem, ul and ll as the upper limit and lower limit, M_c center position.

Retrieval Number: A10531291S419/2019©BEIESP
DOI:10.35940/ijeat.A1053.1291S419

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

K. Hybrid optimization for FJSSP

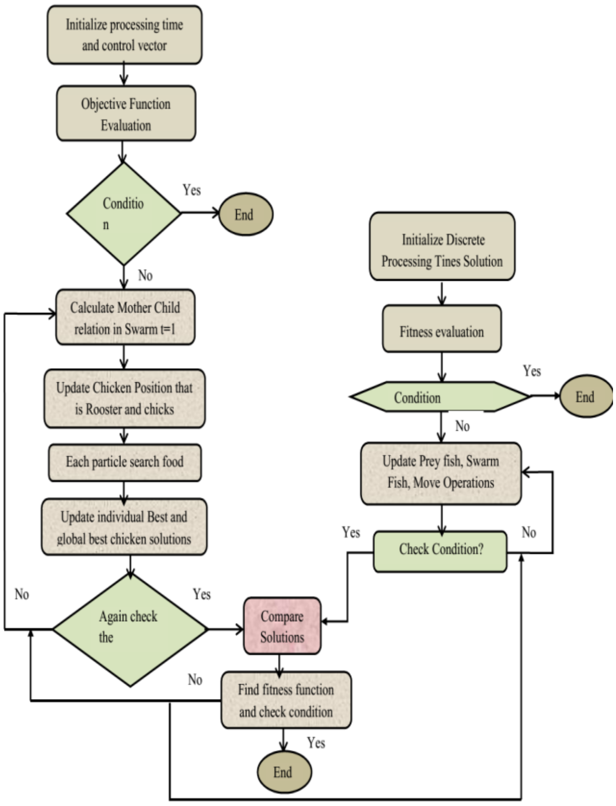


Fig. 3. **Flowchart for the Proposed optimization model**

IV. IMPLEMENTATION OF RESULTS ANALYSIS

This proposed hybrid optimization of FJSSP, implemented in MATLAB 2016a with the system configuration, i5 processors with 4GB RAM. The benchmark data collected from http://people.brunel.ac.uk/~mastjjb/jeb/orlib/, it's having many databases' Validation and the results are discussed in this section. Similarly the other two objectives of FJSSP. For every instance, the minimum workload value noted as an optimal one. It is used definite makespan time to show how this study works properly and how this study result is better than the other studies result in this method.

Therefore, the Gantt char represents the scheduling comparing to the solution in fig 5 and table 3. It also shows the results of MKO7 (20 * 10), processing time like 25, 55, 76 and 58. The number of solution to the problem of 8 by 8 is less, and it also induced the probability of multi – optimal solution is increased. The capacity to avoid local minimum solutions is expanded with the diversity rate increasing. These job elements are processed to provide the minimum makespan time that is added to their corresponding job sequence input time. The number of the machine operated for the problem size varies, and find the processing time.

According to the request of operations, various jobs created the particle position. A particle position is comparing to a machine task all considered operation in light of an alternate priority level relating to an alternative machine

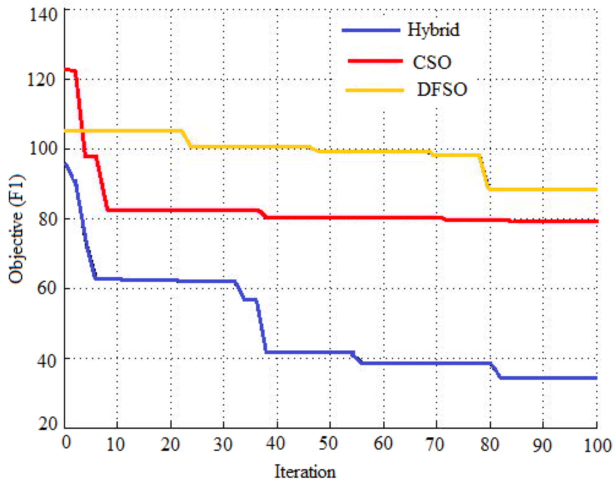
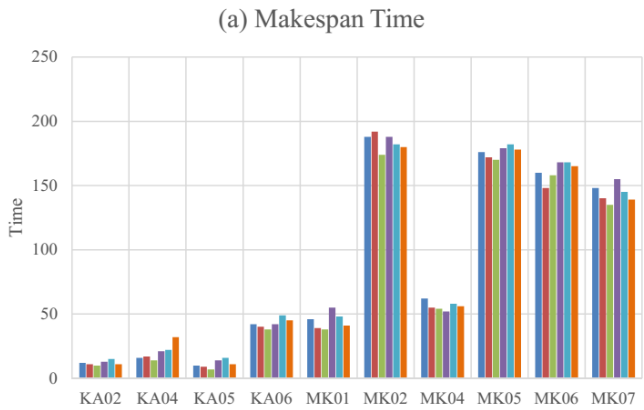


Fig. 4. **Sample Convergence Graph (Time)**

Fish swim promptly in water; truth be told; they are found for food or partners in a wider range. In water, the leave procedure was stopped someplace and the result of AF's behavior will be the same, the distinction between objectives values (food concentration, FC) end up littler within a few cycles. It may fall into local extreme change the parameters haphazardly to the still states for leaping outflow state (28). These operations of anomalous are presented in condition (16) Swarm makes a fish limit in local extreme qualities deploy toward a couple of fish focus on the global value that brings about AF emerging from the local extreme values.

The rate of convergence of the benchmark problem (MKO2) presented in figure 5 with three optimization models which goals to attain the optimal solutions. One to five experiments for various iteration from the range and it is invented that the vast majority of the problem gave great outcomes. During repeated iterations, the convergence graph can situate the optimal fitness value. The close optimal solution with relatively small makes on value is more possible. The hybrid optimization achieved by the optimal fitness value, for instance, minimal value contrasted with CSO and DFSO in the proposed method.



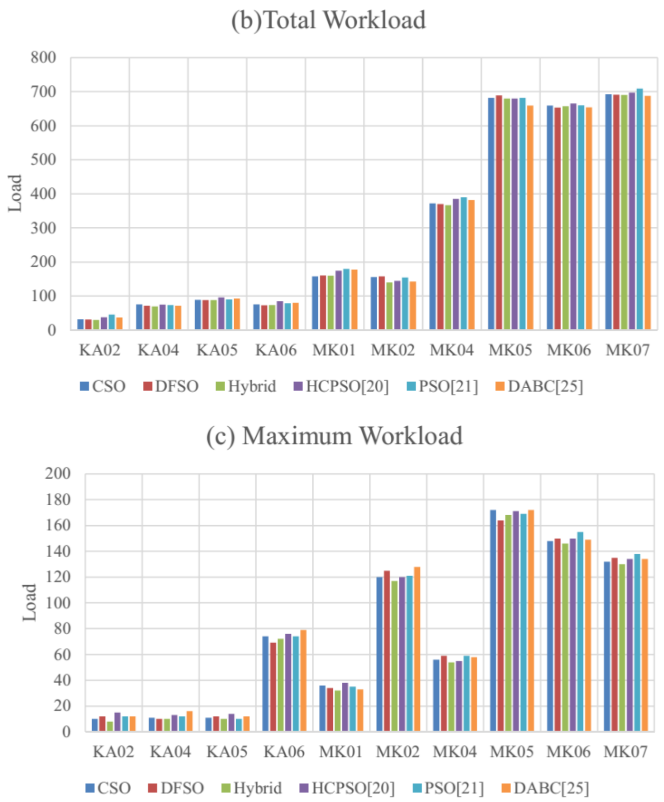


Fig. 5. Comparative Analysis

In fig 4 (a), (b) and (c) presented the makespan time, total work load and maximum workload comparative analysis. Hybrid optimization, CSO, DFSO, HCPSO [15], PSO [16] obtained non – dominated solutions and the compared algorithm dominated the DABC [18] algorithm. (a) Shows the makespan time, it's compared to all optimization with all instance averagely minimum in our proposed work.

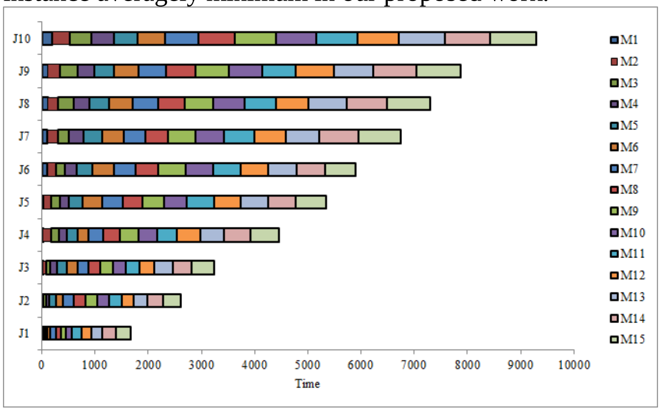


Fig. 6. Gantt chart

The scheduling is comparing to the Solution, therefore represented by the Gantt chart in figure 5 and table 3. Here showing the results of MK07 (20x10), processing time like 25, 55, 76 and 58. There are a number of solutions to the problem of 8 by 8 is less, it also induced the probability of multi-optimal solution to increase. The rate increases with the diversity, the capacity to avoid local minimum solutions are expanded. These job sequences are processed to give the minimum makespan time, which is added to their corresponding job sequence input time. Here, if the problem size varies, the processing time is also found to vary based on the number of the machine operated. Then the particle position can be created stochastically as per the request of operations of various jobs. In light of an alternate priority level relating to an alternate machine, a particle location is comparing to a machine task all considered operations.

V. CONCLUSION

The present paper enhances the flexible job shop scheduling problem with the different benchmark problems. The real procedure is to minimize the makespan with multi-objectives and to effectively overcome the infeasible solutions the objective is developed. So, the guess of the global optimum turns out to be more precisely proportional to the number of iterations. The other one is sometimes local optimum can be happened rather than the global optimum. Multi hybrid optimization investigated with various goals for more than 20 cases. The finest accuracy accomplishes in CSO and DFSO, moreover decreases the makespan time for all machines. . The work can be additionally extended by considering the various exposures which are practiced in a genuine FJSSP problem. . In future research work, might be lessened makespan time with different planning procedures.

APPENDIX

I. OBJECTIVE FUNCTION RESULTS OF BENCHMARK PROBLEMS

Problem	Size J*M	Number of Iterations	F1			F2			F3		
			CSO	DFSO	Hybrid	CSO	DFSO	Hybrid	CSO	DFSO	Hybrid
KA02	4X5	75	12	11	10	32	31	30	10	12	8
KA04	8X8	57	16	17	14	76	72	70	11	10	10
KA05	15X10	59	10	9	7	89	88	88	11	12	10
KA06	10X6	80	42	40	38	76	73	74	74	69	72
MK01	10X6	72	46	39	38	158	161	160	36	34	32
MK02	15X8	56	188	192	174	156	158	140	120	125	117
MK04	15X4	49	62	55	54	372	370	367	56	59	54
MK05	10X15	78	176	172	170	682	689	680	172	164	168
MK06	20X5	62	160	148	158	659	653	657	148	150	146
MK07	20X10	53	148	140	135	692	691	690	132	135	130
Average		65	122	115	114	537	535	533	116	114	113

II. OPERATION TIME FOR MK07 (20x10)

Jobs	Completion Time																			
	M1		M2		M3		M4		M5		M6		M7		M8		M9		M10	
J1	4	O91	6	O51	9	O71	71	O81	71	O11	100	O21	109	O41	158	O61	159	O31	203	O101
J2	4	O12	10	O92	53	O22	140	O62	142	O72	214	O102	215	O32	261	O82	264	O52	339	O42
J3	16	O83	55	O43	57	O53	140	O33	187	O103	218	O73	221	O93	262	O13	355	O23	445	O63
J4	38	O104	57	O74	61	O54	141	O14	268	O24	271	O94	280	O64	280	O34	426	O44	530	O84
J5	40	O15	58	O35	83	O45	144	O75	268	O55	275	O95	306	O65	327	O85	498	O105	544	O25
J6	42	O16	105	O86	105	O76	145	O36	271	O96	279	O56	358	O46	442	O26	504	O106	592	O66
J7	74	O87	107	O57	111	O97	148	O77	303	O67	364	O47	410	O27	442	O37	534	O107	593	O17
J8	75	O38	126	O88	129	O78	152	O98	305	O18	410	O48	442	O68	478	O108	565	O28	593	O58
J9	75	O19	130	O99	170	O89	246	O29	308	O39	436	O109	527	O69	529	O59	641	O49	642	O79
J10	139	O610	139	O510	174	O910	336	O1010	383	O810	521	O210	528	O110	590	O410	644	O710	646	O310
J11	150	O611	171	O1011	178	O711	337	O111	386	O311	523	O911	528	O511	626	O411	722	O211	778	O811
J12	161	O412	199	O612	229	O1012	340	O712	432	O812	525	O112	528	O312	716	O212	723	O512	782	O912
J13	235	O413	320	O213	323	O713	342	O313	465	O1013	614	O813	618	O913	717	O513	723	O113	792	O613
J14	278	O1014	324	O914	422	O814	521	O414	523	O114	614	O314	670	O614	812	O214	813	O514	816	O714
J15	282	O515	393	O615	423	O315	521	O115	584	O415	667	O1015	673	O915	861	O815	863	O715	869	O215
J16	354	O616	395	O216	518	O416	522	O116	587	O516	692	O1016	757	O816	861	O316	867	O716	871	O916
J17	354	O117	396	O517	607	O817	644	O217	657	O417	695	O717	778	O617	863	O317	922	O1017	825	O917
J18	440	O218	528	O618	686	O1018	690	O518	692	O718	696	O318	852	O418	866	O918	970	O818	725	O118
J19	529	O819	603	O1019	697	O619	699	O319	768	O219	768	O519	855	O719	870	O919	971	O119	421	O419
J20	574	O1020	679	O620	701	O920	708	O420	820	O820	821	O320	857	O520	873	O720	971	O120	189	O220

REFERENCES

1. N. Xie and N. Chen,. "Flexible job shop scheduling problem with interval grey processing time", *Applied Soft Computing*, 70, 513-524, 2018.
2. E. Ahmadi, M. Zandieh, M. Farrokh and S.M. Emami " Multi objective optimization approach for flexible job shop scheduling problem under random machine breakdown by evolutionary algorithms", *Computers & Operations Research*, 73, 56-66, 2016.
3. A. Muthiah and R. Rajkumar "A comparison of artificial bee colony algorithm and genetic algorithm to minimize the makespan for job shop scheduling", Procedia Engineering, 97, 1745-1754, 2014.
4. L.N Xing, Y.W. Chen, P. Wang, Zhao, Q.S. and J. Xiong, "A knowledge-based ant colony optimization for flexible job shop scheduling problems", *Applied Soft Computing*, 10(3), 888-896, 2010.
5. A. Thammano, and A. Phu-ang, "A hybrid artificial bee colony algorithm with local search for flexible job-shop scheduling problem", *Procedia computer science*, 20,.96-101, 2013.
6. H.C Chang, Y.P. Chen, T.K Liu and J.H. Chou "Solving the flexible job shop scheduling problem with makespan optimization by using a hybrid Taguchi-genetic algorithm", *IEEE Access*, 3, 1740-1754, 2015.
7. H. Zaher, N. Ragaa and H. Sayed, "A Novel Improved Bat Algorithm for Job Shop Scheduling Problem" *International Journal of Computer Applications*, 164(5), 24-30, 2017.
8. N.P. Hu and W. Pei-li, "An algorithm for solving flexible job shop scheduling problems based on multi-objective particle swarm optimization". *International Symposium on Information Science and Engineering (ISISE)*, 507-511, 2010.
9. C. Turguner and O.K. Sahingoz, "Solving job shop scheduling problem with Ant Colony Optimization", *International Symposium on Computational Intelligence and Informatics (CINTI)*, 385-389, 2014.
10. S. Kavitha, P. Venkumar, N. Rajini, and P. Pitchipoo, "An Efficient Social Spider Optimization for Flexible Job Shop Scheduling Problem", *Journal of Advanced Manufacturing Systems*, 17(02), 181-196, 2018.
11. H. Chen, J. Ihlow, and C. Lehmann, "A genetic algorithm for flexible job-shop scheduling", *International Conference on Robotics and Automation*, Vol. 2, 1120-1125, 1999.
12. C. Peng, Y. Fang, P. Lou, and J. Yan, "Analysis of double-resource flexible job shop scheduling problem based on genetic algorithm", *International Conference on Networking, Sensing and Control (ICNSC)*, 1-6, 2018.
13. T. Jiang, and C. Zhang, "Application of Grey Wolf Optimization for Solving Combinatorial Problems", *Job Shop and Flexible Job Shop Scheduling*, 2018.
14. L. Xu, D. Jiawei, and H. Ming, "Research on Hybrid Cloud Particle Swarm Optimization for Multi-objective Flexible Job Shop Scheduling Problem", 274-278, 2015.
15. S. Huang, N. Tian, Y. Wang, and Z. Ji, "Multi-objective flexible job-shop scheduling problem using modified discrete particle swarm optimization", *Springer Plus*, 5(1), 1432, 2016.
16. N. Kamatchi, and D. Saravanan, "A Hybrid algorithm using firefly and cuckoo search algorithm for flexible open shop scheduling problem" , *Journal of Mechanical and Production Engineering Research and Development*, Vol.7(2), 155-166, 2017.
17. J.Q. Li, Q.K. Pan, and M.F. Tasgetiren, "A discrete artificial bee colony algorithm for the multi-objective flexible job-shop scheduling problem with maintenance activities", *Applied Mathematical Modelling*, 38(3), 1111-1132, 2014.
18. T.K.Dao, T.S. Pan, and J.S. Pan, "Parallel bat algorithm for optimizing makespan in job shop scheduling problems". *Journal of Intelligent Manufacturing*29(2), 451-462, 2018.
19. D. Wu, F. Kong, W. Gao, Y.Shen and Z. Ji, "Improved chicken swarm optimization", *International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, 681-686, 2015.
20. D. Pythaloka, A.T. Wibowo, and M.D. Sulistiyo, "Artificial fish swarm algorithm for job shop scheduling problem", *International Conference on Information and Communication Technology (ICoICT)*, 437-443, 2015.

AUTHORS PROFILE

S. Kavitha, received her Bachelor degree in Mechanical Engineering from Bharathiyar University, Coimbatore, India and Masters in CAD from Sathyabama University, Chennai, India. She is currently pursuing her PhD in Kalasalingam University, Tamilnadu, India. She is also working as an Assistant Professor in the Department of Mechanical Engineering, Kalasalingam University. Her research interests include operations management and scheduling.

Dr. P. Venkumar, is currently a Senior Professor in Department of Mechanical Engineering, Kalasalingam Academy of Research and Education, Krishnankovil, Tamilnadu, Inida. He received his B.E., and M.E., degrees in Thigagrajar College of Engineering, Madurai, Tamilnadu, India, in 1994 and 1997, respectively and Ph.D. degree in Manonmaniam Sundaranar University, India in Industrial Engineering in 2006. His main areas of research interest are Manufacturing system optimization, Supply chain management and Smart manufacturing.