A Hybrid Firefly Algorithm for Multi Objective Optimization in Flexible Manufacturing System

Gayathri Devi K[1], RS Mishra[2], AK Madan[3]

[1]Research Scholar, Department of Mechanical and Production Industrial and Automobiles Engineering, Delhi Technological University, Delhi, India
[2,3]Professor, Department of Mechanical and Production Industrial and Automobiles Engineering, Delhi Technological University, Delhi, India

---***---

Abstract – *Manufacturing industries are undergoing tremendous change due to high demands in market and globalization. These demands are forcing the companies to adapt to flexibility. And Flexible job shop problem is one such change that is gaining importance in current scenario. In this paper a hybrid technique named (HyFA) is proposed which is combination of firefly with simulated annealing and Greedy search method to solve a multi objective problem of minimizing the idle time of the machine and minimizing the total penalty cost for not meeting the deadline concurrently. The algorithm has been applied to a problem taken from literature and the results were found to be conclusive.*

Key Words: Job shop scheduling, Firefly algorithm, simulated annealing, greedy search, meta-heuristics, Optimization.

1. INTRODUCTION

An important application area in Flexible Manufacturing Systems *(FMSs)* is machine scheduling. The machine scheduling gained importance due to the chance of improving efficiency of a job shop while its flexibility is maintained. Jobs might have due dates or deadlines, release times, or weights indicating their relative importance. The possibilities of part processing might be restricted by certain precedence constraints, or each operation (i.e. task) can be carried out independently of the others which are necessary for part completion. Objectives might consist of minimizing schedule length, mean flow time or due date involving criteria. All these problem characteristics are well known from traditional machine scheduling theory, and had been discussed earlier. Most of the FMS-scheduling problems have to take into account these problem formulations which lead to more complex computation hence the problem becomes NP hard. The primary goal of any manufacturing industry is to achieve a high level of productivity and flexibility, which can only be done in a fully integrated manufacturing environment. A flexible manufacturing system (FMS) is an integrated computer-controlled configuration in which there is some amount of flexibility that allows the system to react in the case of changes, whether predicted or unpredicted. FMS consists of three main systems. The work machines that are often automated CNC machines are connected by a material handling system (MHS) to optimize parts flow and the central control computer, which controls material movements and machine flow. An FMS is modeled as a collection of workstations and automated guided vehicles (AGV). It is designed to simultaneously manufacture a low to medium volumes of a wide variety of high quality products at low cost. The flexibility is generally considered to fall into two categories, which both contain numerous sub-categories. The first category, machine flexibility covers the system's ability to be changed to produce new product types, and ability to change the order of operations executed on a part. The second category is called routing flexibility, which consists of the ability to use multiple machines to perform the same operation on a part, as well as the system's ability to absorb large-scale changes, such as in volume, capacity, or capability. The main objective of an automated plant is to efficiently schedule the production process based upon some computer fed logics. Various algorithms have been used to develop for the decision making and scheduling processes in FMS. In recent years, hybridization of non-traditional algorithms are gaining popularity and they tend to give better results than an individual technique. Hence an attempt has been made to combine Firefly algorithm, Simulated annealing, greedy heuristics leading to a novel hybrid technique named HyFA that has been used to address the decision making problems. Scheduling is the process of generating the schedule and schedule is a physical document and generally tells the happening of things and shows a plan for the timing of certain activities. Generally, scheduling problem can be approached in two steps; in the first step sequence is planned or decided how to choose the next task. In the second step, planning of start time and perhaps the completion time of each task is performed. In a scheduling process, the type and amount of each resource should be known so that accomplishing of tasks can be feasibly determined. Boundary of scheduling problem can be efficiently determined if resources are specified. In addition, each task is described in terms of such information as its resource requirement, its duration, the earliest time at which it may start and the time at which it is due to complete.

2. LITERATURE REVIEW

The optimization of scheduling are usually done by different approaches spanning from mathematical approaches and branch and bound techniques to bottleneck based heuristics, artificial intelligence and by local search methods. This section of literature helped the research scholar in exploring the FMS and to understand the need for better scheduling. The summary of literature surveyed in this regard is given below.

Chin-Chia Wuet al [1]addressed a single-machine total completion time problem with learning effect and release times based on the sum of processing times. A simulated-annealing algorithm was also proposed to obtain a near optimal solution. They proposed for Future research may consider other criterion such as the tardiness or lateness or studying the problem in the multi-machine setting. A.Noorul Haq, T. Karthikeyan and M.Dinesh [2] proposed and integrated scheduling of FMS, which is conforming to material handling system. First they employed a Giffler and Thompson heuristic with six priority-dispatching rules for determining the best routing of the jobs and this optimum part routing is considered as an input to schedule the material handling system. In order to minimize the distance travelled by the AGV and transportation cost of the AGV, a hybrid approach of GA and simulated annealing is employed for obtaining quality solution. Jerald.J, Asokan.P, Prabaharan.G, Saravanan.R [3] presented different scheduling mechanisms to generate an optimum schedule based on the combination of both traditional and non-traditional approaches. They formulated a multi objective function to minimize the idleness and penalty and optimized it by using different optimization approaches such as GA, SA, Mematic algorithm and PSOA. They compared the results of each of these approaches and presented the results. For the purpose of illustration, they considered 43 jobs, 16 machines system and further considered two more types of problems such as 10 jobs, 8 machines and 20 jobs, 15 machines. The results show that PSOA is found to be better than other three algorithms and provides the minimum combined objective function value.

S.G.Ponnambalam, V.Ganapathy, S.Saravana Sankar and R.Karthikeyan, (2002)[4] proposed a Tabu search based method to generate optimum schedule for a 43 jobs, 16 machines FMS system by considering dual objectives such as the minimum total penalty cost and the minimum machine idleness. They compared the results of scheduling achieved using Tabu Search with other priority rules like LPT, SPT, LBQ, SBQ, HP and EDD. They also demonstrated that the Tabu search based method is capable of better performance even when compared to the SPT based scheduling rule.

Veeranna.V [5] emphasized the need for distributing different resources to achieve the maximum efficiency in FMS layout. It has been estimated that the material handling cost involves 15% to70% of the manufacturing costs. The results of the optimal design of the physical layout indicate that efficient arrangement of devices reduces manufacturing costs to an extent of 10% to 30 %. Yumin He et al. [6]proposed a state-dependent algorithm for the FMS robot scheduling problem in make-to-order environments for mass customization. The performance of the proposed algorithm is compared with an effective FMS robot scheduling rule and the shortest remaining processing time first rule. From the results, it is concluded from the results, that the effectiveness of the proposed algorithm enhances the productivity of the FMS. They also discussed the practical application and also provided further the scope for future research.

M. Bank, S.M.T. Fatemi Ghomi, F. Jolai, J. Behnamian [7]considered a permutation flow shop scheduling problem with deteriorating jobs. A particle swarm optimization algorithm with and without a proposed local search was developed to determine a job sequence which minimizes the total tardiness criterion. Furthermore, a simulated annealing algorithm was pro- posed to solve the problem. They found that PSO gives more promising results than the simulated Annealing(SA). Future scope is for multiobjective and trying other new metaheuristics techniques. In Narendhar. S and Amudha. T [8]research work, they have proposed a new hybrid technique of Bacterial Foraging Optimization with Ant Colony Optimization named Hybrid Bacterial Foraging Optimization for solving Job Shop Scheduling Problem. They solved for single objective of minimising makespan for Admas, Balas and Zawaxk (ABZ), Lawrence (LA) Benchmark problems. Shashikant Burnwal & Sankha Deb [9] minimizing the penalty cost due to delay in manufacturing and maximizing the machine utilization time.(COF) using cuckoo search. A study by Enzhou [10] proposed a multi-agent system for scheduling with for flexible manufacturing system(FMS). However, this research failed to provide a contribution of PSO on schedule since the developed model adopts both TS and PSO for optimization. A study by Kumar et al.[11] [5] investigated the scheduling problem associated with FMS by the application of metaheuristics approach improvement in planning for production for FMS scheduling. Scheduling for optimization involves Bacterial Foraging optimization algorithm (BFOA, GA and Differential Evolution (DE) for an optimal scenario in to consideration. Gaurav Kumar[12] used Genetic Algorithm (GA) and Taguchi L27 array to maximize system utilization and throughput. Yunqiang Yin et al[1] proposed honey-bees optimization algorithm+Branch and bound method to minimize the weighted sum of the completion times of the jobs. Xinyu Li, Liang Gao [13] minimized makespan by developing hybrid genetic algorithm and tabu search. Win-Chin Lina et al [14]developed particle swarm optimization algorithm (PSO), an opposite-based particle swarm optimization (OPSO) algorithm, and a particle swarm optimization algorithm with a linearly decreasing inertia weight (WPSO) to to minimize the total completion time of the orders of one agent, with the restriction that the total completion time of the orders of the other agent cannot exceed a given limit. V.K. Chawlaa, A. K. Chanda and Surjit Angra[15] developed grey wolf optimization algorithm (GWO) to balance the workload of AGVs and to minimize the travel time of AGVs.Nidhish Mathew et al [16] considered a FMS that has 32 CNC Machine tools for processing 40 varieties of products. They solved it for minimizing machine idle time and minimizing total penalty cost which are contradictory objectives. They have developed a multi-objective optimization procedure based on NSGA-II and software has been developed using .net programming for setting the optimum product sequence. A Global–optimal front was then obtained using the software after 3000 generations.RS Mishra et al[17] have done a detailed review on AGV's. They have studied various approaches that have been adapted to schedule FMSs like simulation and analytical methods. In their paper, the literature dealing with the parts scheduling problem in flexible manufacturing systems (FMS) has been reviewed. The book chapter by Lindfield[18] has given a detailed description of forefly algorithms which helped us to get a better understanding of how the algorithm works and the adaptations that can be followed.

3. PROBLEM DESCRIPTION

The problem environment, assumptions and aim of the present work are as follows [3, 19, 20]

The FMS, which is considered in this work, has a configuration as shown in figure 1. There are four flexible machining cells (FMCs), with three to six computer numerical machine centers (CNCs). Each one is provided with independent tool magazines, part program controller, automatic tool changer (ATC) and buffer storage, part-carrying conveyors, a robot and an automated storage and retrieval system which are linked by means of host computer. The four FMCs are connected by automated guided vehicles (AGV). These are used to transport material from loading to unloading station. These AGVs perform the inter- cell movements between the FMCs, movement of loaded pallets from the loading station to any of the FMCs, movement of finished product from any of the FMCs to the unloading station and the movement of semi finished product between the AS/RS and the FMCs. There is a loading station from where the parts are released in batches and unloading station from where the finished parts are collected.

- There is one automatic storage and retrieval system (AS/RS) to store the work in progress.
- The five FMCs are connected by two identical automated guided vehicles (AGVs). These AGVs perform the inter- cell movements between the FMCs, the movement of finished product from any of the FMCs to the unloading station and the movement of semi-finished products between the AS/RS and the FMCs.
- There is a dedicated robot for loading and unloading AGVs.

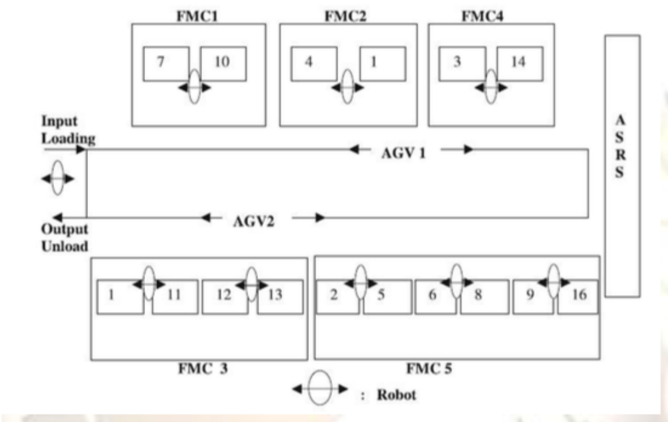


Figure 1: FMS structure

The primary objective of this research is,

- To develop a multi-objective mathematical model towards enhancing the performance
- To minimize the idle time of the machine and to minimize the total penalty cost.

Combined objective function (COF)

$$COF = (w1) * \frac{\text{Total penalty cost}}{\text{Maximum permissible penalty}} + (w2) * \frac{\text{Total machine idle time}}{\text{Maximum total machine elapsed time}}$$

where w1and w2 are the weights assigned to each objective function. (Equal weights w1 = 0.5and w2 = 0.5 are given in the experiment conducted)

3.1 Assumptions

The assumptions made in this work are as follows:

There are 40 to 50 varieties of products for a particular combination of tools in the tool magazines.
Each type/variety has a particular processing sequence batch size, deadline and penalty cost for not meeting the deadline.
Each processing step has a processing time with a specific machine.

4. PROPOSED METHODOLOGY

In this study, we developed an effective multi-objective scheduling approach which is a combination of the Firefly, simulated annealing and greedy heuristic approach.

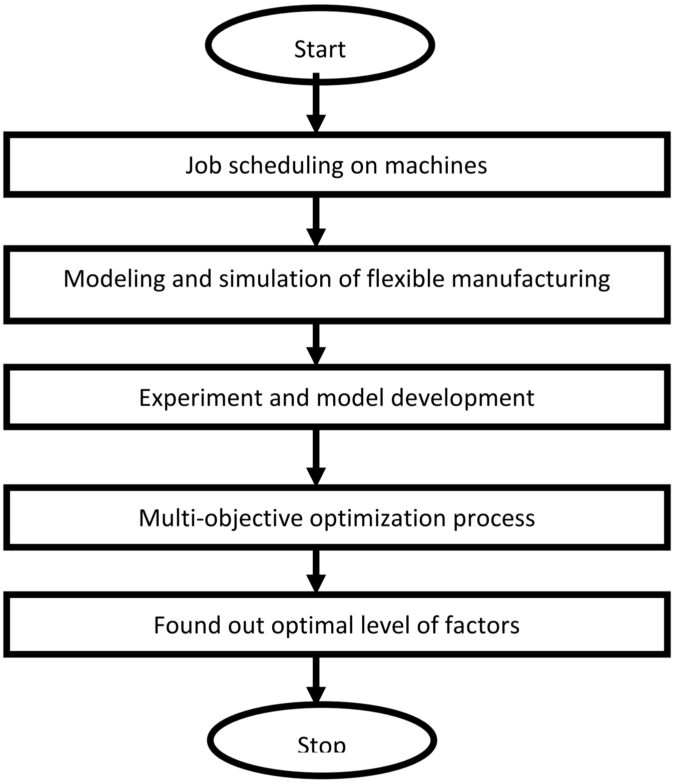


Figure 2 Proposed Research Approach

The purpose of using a multi-objective genetic algorithm for effectively solving multistage-based processing schedules in FMS environment. By the use of simulated annealing can locate a good approximation to the global minimum of a given function in large search space. It is used when the search space is discrete. Then the greedy heuristics approach is used to solve precisely the sequencing issues. The proposed algorithms have been implemented and tested by using Matlab R2016b, computing environment on an Intel Core™i7, with Windows 10.

The pictorial representation of the proposed research approach and flow of the process is shown in Figure 2. First, it starts with the job scheduling via sequencing rule process. Then will develop the scheduling of simulated flexible manufacturing process. And finally find the optimal solutions for multi-objective optimization process.

4.1 Firefly Algorithm(FA)

The Firefly Algorithm (FA) is a nature - inspired algorithm which is based on the social flashing behavior of fireflies. This algorithm was proposed by Xin-She Yang[21] in 2008. A significant advantage of the algorithm is the fact that it uses mainly real random numbers, and it is based on the global communication among the swarming particles i.e., the fireflies, and as a result, it seems more effective multi objective optimization. In this algorithm, the flashing light helps fireflies for finding mates, attracting their potential prey and protecting themselves from their predators. The swarm of fireflies will move to brighter and more attractive locations by the flashing light intensity that associated with the objective function of problem considered in order to obtain efficient optimal solutions. Attractiveness of a firefly is proportional to its brightness and for any couple of fireflies, the brighter one will attract the other; so the less bright one is moved towards the brighter one. This is performed for any binary combination of fireflies in the population, on every iteration of algorithm. Firefly algorithm idealizes some of the characteristics of the firefly behavior. They follow three rules: a) all the fireflies are unisex, b) each firefly is attracted only to the fireflies ,that are brighter than itself; Strength of the attractiveness is proportional to the firefly's brightness ,which attenuates over the distance ; the brightest firefly moves randomly and, c) brightness of every firefly determines it's quality of solution ; in most of the cases, it can be proportional to the objective function. Using the above three rules, a pseudo-code of the Firefly Algorithm is given in Figure 3

```
Objective function f(x), x = (x_1, ..., x_d)^T.
Generate an initial population of n fireflies x_i (i = 1, 2, ..., n).
Light intensity I_i at x_i is determined by f(x_i).
Define light absorption coefficient γ.
while (t <MaxGeneration),
for i = 1 : n all n fireflies
   for j = 1 : n all n fireflies (inner loop)
   if (I_i < I_j)
      Move firefly i towards j.
      end if
      Vary attractiveness with distance r via exp[−γr^2].
      Evaluate new solutions and update light intensity.
   end for j
end for i
Rank the fireflies and find the current global best g_*.
end while
Post-process results and visualization.
```

Figure 3 Pseudo code of Firefly Algorithm

In the above algorithm , m is the number of the fireflies, I_0 is the light intensity at the source, ϒ is the absorption coefficient and α is the size of the random step . All these parameters will be explained further in detail.

Distance (r): The distance between two generic fireflies (e.g. i and j) is expressed by the Cartesian distance as follows

$$r_{ij} = \|pi - pj\| = \sqrt{\sum_{k=1}^{n} (p_{ik} - p_{jk})^2}$$

Where p_i, p_j are the spatial co-ordinates

Intensity I(r): In the traditional FA the light intensity perceived of each firefly depends on the distance r and it is proportional to the value of the objective function. In line with the formulation provided by Yang (2009), it can be expressed as:

$$I = I_o e^{\gamma r^2}$$

Here, I_o - original light intensity

γ - Coefficient of light absorption.

Attractiveness (β): This represents a relative measure of the light perceived by beholders and other fireflies. Therefore the formulation of the attractiveness is expressed as:

$$\beta = \beta_o e^{-\gamma r^2}$$

Here, β_o denotes the attractiveness when the distance r is equal to zero.

Movement: The movement of the generic firefly i, that is attracted by a brighter one, is expressed as:

$$p_i(t+1) = p_i + \beta_o e^{-\gamma r^2 ij}(p_i(t) - p_i(t)^2) + \alpha\epsilon_i$$

In this equation $p_i(t)$ represents the position of the firefly at the time t; $\beta_o e^{-\gamma r^2 ij}(p_i(t) - p_i(t)^2)$ represents the attraction between fireflies; $\alpha\epsilon_i$ represents the randomness of the process where the vector ϵ_i includes random numbers extracted from a normal distribution and α is a random parameter.

4.2 Simulated Annealing

Simulated Annealing is a stochastic computational method for finding global extremisms to large optimization problems. Kirkpatrick[22] in 1983 and Cerny in 1984 first proposed it as an optimization technique.

Simulated Annealing (SA) algorithm is a nature- inspired method which is adapted from process of gradual cooling of metal in nature.

In the metallurgical annealing process, a solid is melted at high temperature until all molecules can move about freely and then a cooling process is performed until thermal mobility is lost. The perfect crystal is the one in which all atoms are arranged in a low level pattern, so crystal reaches the minimum energy. A classical simulated annealing begins by generating an initial solution randomly. At each stage, the new solution taken from the neighborhood of the current solution is accepted as the new current solution if it has a lower or equal cost; if it has a higher cost it is accepted with a probability that decreases as the difference in the costs increases and as the temperature of the method decreases. This temperature, which is simply a positive number, is periodically reduced by some temperature scheme, so that it moves gradually from a relatively high value to lower value as the method progresses. Thus at the start of SA, most worsening moves are accepted, but at the end only improving ones are likely to be accepted. The method converges to a local optimum as the temperature approaches zero, because SA has performed many perturbations at a high temperature which have pushed the search path into new areas, a better local optimum solution should hopefully be reached.

4.2.1 Parameters of Hybrid Simulated Annealing

- **Initial Temperature:** The temperature is the controlled parameter in simulated annealing that decreased steadily as the algorithm proceeds. It determines the probability of accepting an inferior solution at any step and is used to bound the extent of the search in a given dimension. At very high temperature (T), the SA algorithm should have equal chance of rejecting or accepting a worse solution. This important requirement of a successful SA has been disregarded by past SA research which tends to set T at a value exclusively based on available computer processing time and number of variables. So for fair comparison its value has been fixed which is equal to number of jobs.

- **Annealing Function:** The annealing function modified current schedule and return a new schedule that has been changed by an amount proportional to the temperature. The new schedule is produced by randomly swapping two points in a schedule.

- **Temperature Function:** It updates the temperature vector for annealing process. In this work, exponentially temperature reduction is applied in which temperature at iteration is reduced exponentially at each iteration.

Annealing rate directly affects the speed and accuracy of the annealing method. The general annealing rate function is measured by

$$T(t) = \beta \times T(t-1)$$

Where $\beta \in (0, 1)$ - annealing rate coefficient;

T(t) and T(t − 1) - current temperature and previous temperature,

t - Iteration number.

4.3. Greedy Heuristics

The greedy algorithm makes a locally optimal solution. That is it tries to find local optimal solution at each stage with the intent to find a global optimal solution. In many problems, a greedy strategy does not in general produce an optimal solution, but nonetheless a greedy heuristic may yield locally optimal solutions that approximate a global optimal solution in a reasonable time .In a greedy algorithm, the optimal solution is built up one piece at a time. At each stage the best feasible candidate is chosen as the next piece of the solution. There is no backtracking

4.3.1 Specifics:

- A candidate set, from which a solution is created
- A selection function, which chooses the best candidate to be added to the solution
- A feasibility function that is used to determine if a candidate can be used to contribute to a solution
- An objective function which assigns a value to a solution, or a partial solution
- A solution function, which will indicate when we have discovered a complete solution

4.4 Optimization Procedure

4.4.1 Matlab Programming

Matlab software is used for programming the proposed approach. The use of matlab enables us to solve complex scheduling problems involving different job types and multiple machines. The program is coded in such a way that the user can have flexibility in terms of varying the number of machines and number of jobs. The input to the program is given in the form of data stored in Excel sheet Table-1. The data includes machine timings for each operations, operation sequence, batch size, and penalty and due date. Parameters used in this experiment are given in Table 2.

Table 1. *Machining sequence, P.T- process time (in min),D.D (due date in days),B.S(batch size in No's) and P.C (penalty cost in Rs/units/day)(43jobs-16machines)*

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15	M16	B.S	P.C	D.D
P1	0	0	0	0	0	1	1	1	0	2	0	0	0	0	0	0	150	1	17
P2	0	1	0	0	0	1	0	2	2	0	0	0	0	4	0	2	200	1	17
P3	0	0	0	0	0	0	0	1	0	0	3	0	4	0	0	0	800	1	14
P4	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	700	2	26
P5	0	0	0	5	3	0	0	0	0	0	0	0	0	0	4	0	150	1	11
P6	0	0	0	0	0	5	0	0	0	0	0	0	0	1	0	0	700	1	16
P7	0	0	5	0	0	3	0	0	0	0	0	0	0	0	0	5	250	2	26
P8	0	0	0	0	4	5	0	1	0	0	0	0	0	0	0	0	850	2	26
P9	0	0	0	1	5	0	0	1	0	0	1	0	0	0	0	0	100	0	1
P10	0	2	0	0	0	0	0	0	1	0	0	0	0	0	0	4	150	2	20
P11	0	0	0	0	0	0	0	4	0	0	0	2	0	0	0	0	250	1	1
P12	0	0	0	0	0	2	0	4	0	1	0	0	0	0	0	0	1000	3	19
P13	0	0	0	0	0	1	5	0	0	4	0	0	0	0	0	0	700	4	25
P14	0	0	0	2	3	2	0	0	0	0	0	0	0	0	2	0	1000	4	22
P15	0	0	0	0	4	0	0	3	0	0	0	0	0	0	0	0	700	5	15
P16	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	750	3	27
P17	0	0	1	0	0	4	0	0	0	0	0	0	0	1	0	0	650	4	20
P18	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	3	250	5	24
P19	0	0	0	1	5	2	0	2	0	0	0	0	0	0	5	0	450	1	5
P20	0	0	0	0	0	0	0	2	0	0	4	0	0	0	0	0	50	5	11
P21	0	0	0	5	5	0	0	4	0	0	0	0	0	0	4	0	850	3	16
P22	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	200	5	24
P23	0	0	0	2	1	5	0	4	0	0	0	0	0	0	0	0	50	4	14
P24	0	0	0	0	0	0	0	4	0	0	4	5	4	0	0	0	200	5	7
P25	0	0	0	0	0	0	3	0	0	2	0	0	0	0	0	0	350	1	24
P26	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	450	0	27
P27	0	0	0	0	0	0	0	5	0	0	5	4	0	0	0	0	400	1	22
P28	0	1	0	0	0	0	0	1	2	0	0	0	0	0	0	0	950	5	3
P29	0	0	0	1	5	0	0	0	0	0	0	0	0	0	0	0	700	1	7
P30	0	0	0	0	0	0	0	0	0	0	3	5	0	0	0	0	1000	1	18
P31	0	0	0	0	0	0	0	2	0	2	0	0	0	0	0	0	800	2	2
P32	0	3	0	0	0	4	0	0	3	0	0	0	0	0	0	0	800	1	15
P33	0	0	0	0	4	5	0	0	0	0	0	0	0	0	3	0	500	4	27
P34	0	0	2	0	0	2	0	0	0	0	0	0	0	0	0	0	300	4	12
P35	0	0	4	0	0	0	0	0	0	0	0	0	0	1	0	0	900	2	9
P36	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	700	2	20
P37	5	2	0	0	0	3	0	3	2	0	0	0	0	0	0	4	250	4	22
P38	0	4	0	0	0	0	0	3	2	0	0	0	0	0	5	0	50	1	8
P39	0	0	0	0	0	5	0	0	0	5	0	0	0	0	0	0	500	1	9
P40	0	2	0	0	0	4	0	0	4	0	0	0	0	0	0	0	250	5	7
P41	0	0	0	0	1	0	0	2	0	0	0	0	0	0	1	0	800	4	22
P42	0	5	0	0	0	4	0	0	3	0	0	0	0	0	0	1	400	2	19
P43	3	0	0	0	2	2	0	2	0	0	0	0	0	0	3	0	550	3	15

The implementation process flow is discussed as follows:

Step 1: Initialize random population within predefined boundary

Step 2: Evaluate fitness of each individual machine

Step 3: Apply mutation to the current state and Generate neighbor solutions;

Step 4: calculate fitness of the current state and mutated solutions on the basis of intensity and absorption coefficient, attractiveness

Step 5: Generate solution current state in the neighborhood of current state

Step 6: compute cost and task completion time

Step 7: perform step 2 to 7 until a termination condition is satisfied

Here the termination condition is user defined of maximum generations of 200 or when there is no considerable improvement in the fitness function.

Table 2:Parameters of Hybrid Firefly (HyFa)

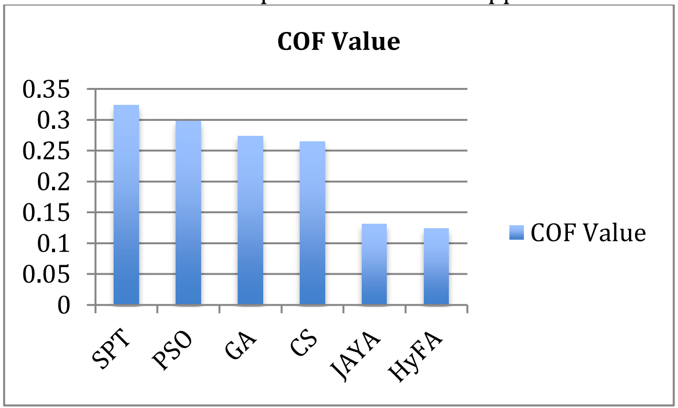
Parameter	Value
firefly size in population (D)	30
Annealing function (B)	Fast annealing
Initial temperature and annealing function (T)	100
The total number of generations, (k)	200
Initial attractiveness between two fireflies (β_0)	0.1
Absorption coefficient of the least intensity firefly (γ)	1
The maximum iteration size (i)	800
Randomization parameters (α)	0.5

5. RESULTS AND DISCUSSIONS

The optimization procedures developed in this work are based on the various non-traditional approaches that have been implemented using Matlab 2016b and above. Different optimal schedules for the given problem are taken from literature and compared. Among the approaches found in the literature, the schedule obtained by the Hybrid Firefly algorithm(HyFA) give the optimal COF value, i.e. minimum total penalty cost and minimum machine idle time as shown in the Table 3.

Combined objective function obtained is **0.1240752.**
Optimum sequence:
23,28,1,38,9,30,9,3,25,13,31,34,24,7,40,26,5,4,27,2, 20,16,10,36,18,11,37,8,19,12,41,22,42,29,32,15,43,1 7,21,6,33,14,35.

Table 3 Comparison of Various approaches



6. CONCLUSION

A novel hybrid technique of Firefly (HyFA) has been implemented successfully for solving the scheduling optimization problem of FMS. Results are obtained for the 43 jobs and 16 machines FMS system. With less computational effort it is possible to obtain the solution for such a large number of jobs (43) and machines (16). This work leads to the conclusion that the procedures developed in this work can be suitably modified to any kind of FMS with a large number of components and machines subject to multi objective functions. Future work will include availability and handling times of loading/unloading stations, robots and AGVs.

REFERENCES:

1. Yin Y, Wu WH, Cheng TCE, et al (2015) A honey-bees optimization algorithm for a two-agent single-machine scheduling problem with ready times. Appl Math Model 39:2587–2601. https://doi.org/10.1016/j.apm.2014.10.044
2. Noorul Haq A, Karthikeyan T, Dinesh M (2003) Scheduling decisions in FMS using a heuristic approach. Int J Adv Manuf Technol 22:374–379. https://doi.org/10.1007/s00170-002-1474-0
3. Jerald J, Asokan P, Prabaharan G, Saravanan R (2005) Scheduling optimisation of flexible manufacturing systems using particle swarm optimisation algorithm. Int J Adv Manuf Technol 25:964–971. https://doi.org/10.1007/s00170-003-1933-2
4. Sankar SS, Ponnambalam SG (2004) An intelligent integrated scheduling model for flexible manufacturing system. 1100
5. Sreedhar Kumar AVS, Veeranna V, Durgaprasad B (2015) Multi Objective Scheduling of Jobs in Flexible Manufacturing System Using Metaheuristic Approaches with Inclusion of Simulation Modeling. Int J Comput Eng Res Trends 351:2349–7084
6. He Y, Stecke KE, Smith ML (2016) Robot and machine scheduling with state-dependent part input sequencing in flexible manufacturing systems. Int J Prod Res 54:6736–6746. https://doi.org/10.1080/00207543.2016.1161252
7. Behnamian J, Fatemi Ghomi SMT (2014) Multi-objective fuzzy multiprocessor flowshop scheduling. Appl Soft Comput J 21:139–148. https://doi.org/10.1016/j.asoc.2014.03.031
8. Narendhar S, Amudha T (2012) A Hybrid Bacterial Foraging Algorithm For Solving Job Shop Scheduling Problems. Int J Program Lang Appl 2:1–11
9. Burnwal S, Deb S (2013) Scheduling optimization of flexible manufacturing system using cuckoo search-based approach. Int J Adv Manuf Technol 64:951–959. https://doi.org/10.1007/s00170-012-4061-z
10. Enzhou H (2014) Particle swarm optimization combined with tabu search in multi-agent scheduling for FMS. In: J. Liaoning Tech. Univ. Sci.
11. Kumar AVSS, Veeranna V, Durgaprasad B, Sarma BD (2013) Combined Objective Optimization Of FMS Scheduling With Non-Traditional Optimization Techniques. Int J Eng Res Technol (IJERT 2:1420–1428
12. Kumar G (2015) The Simulation Optimization of a Flexible Manufacturing System with Arena. 4:755–760
13. Lu C, Li X, Gao L, et al (2017) An effective multi-objective discrete virus optimization algorithm for flexible job-shop scheduling problem with controllable processing times. Comput Ind Eng 104:156–174. https://doi.org/10.1016/j.cie.2016.12.020
14. Lin WC, Yin Y, Cheng SR, et al (2017) Particle swarm optimization and opposite-based particle swarm optimization for two-agent multi-facility customer order scheduling with ready times. Appl Soft Comput J 52:877–884. https://doi.org/10.1016/j.asoc.2016.09.038
15. Chawla VK, Chanda AK, Angra S (2019) The scheduling of automatic guided vehicles for the workload balancing and travel time minimi-zation in the flexible manufacturing system by the nature-inspired algorithm. J Proj Manag 4:19–30. https://doi.org/10.5267/j.jpm.2018.8.001
16. Mathew N, Saravanan NR (2012) Evaluation of Genetic Algorithm Approach for Scheduling Optimization of Flexible Manufacturing Systems. 2:437–446
17. Mishra R, Devi GK, Madan A (2019) FMS scheduling using AGVs by soft computing techniques: A Review. Int J Res Eng Innov 3:200–207
18. Lindfield G, Penny J (2017) The Firefly Algorithm. In: Introduction to Nature-Inspired Optimization. Elsevier, pp 85–100
19. Jawahar N, Aravindan P, Ponnambalam SG, Suresh RK (1998) AGV schedule integrated with production in flexible manufacturing systems. Int J Adv Manuf Technol 14:428–440.

https://doi.org/10.1007/BF01304622
20. Gnanavel Babu A, Jerald J, Noorul Haq A, et al (2010) Scheduling of machines and automated guided vehicles in FMS using differential evolution. Int J Prod Res 48:4683–4699. https://doi.org/10.1080/00207540903049407
21. Yang X (2010) Nature-Inspired Metaheuristic Algorithms Second Edition
22. Udaiyakumar KC, Chandrasekaran M (2014) Optimization of multi objective job shop scheduling problems using firefly algorithm. Appl Mech Mater 591:157–162. https://doi.org/10.4028/www.scientific.net/AMM.591.157