International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** XII **Month of publication:** December 2021

DOI: https://doi.org/10.22214/ijraset.2021.39536

A Hybrid Firefly Algorithm Approach for Job Shop Scheduling Problem

Gayathri Devi K [1], R. S. Mishra [2], A. K. Madan [3]
[1]*Research Scholar,* [2]*Professor & Dean,* [3]*Professor, Department of Mechanical and Production Industrial and Automobiles Engineering, Delhi Technological University, Delhi, India*

Abstract: Job shop scheduling has always been one of the most sought out research problems in Combinatorial optimization. Job Shop Scheduling problems (JSSP) are categorized under NP hard problems. In recent years the meta heuristic algorithms have been proved effective to solve hardcore NP problem. Firefly Algorithm is one of such meta heuristic techniques which is nature inspired from firefly characteristic. Its potential can be enhanced further by hybridizing it with other known evolutionary algorithms and thereby getting improved results in less computational time. In this paper we have proposed a new hybrid technique christened as HyFA, by hybridizing Firefly algorithm(FA) with simulated annealing (SA) and Greedy heuristics approach (GHA). The hybrid technique has the advantages of all three algorithms and are combined in such a way that a quicker and better optimal solution is obtained. Our proposed HyFA is coded in Matlab with an objective to minimize the makespan (C_m). The novel hybrid technique is then used to evaluate 1-25 Lawrence problems taken from literature. The results show the proposed technique is more effective not only in getting optimal results but has significantly reduced computational time.

Keywords: Scheduling, Optimisation, Job shop scheduling, Meta-heuristics, Firefly, Simulated Annealing, Greedy heuristics Approach.

I. INTRODUCTION

Scheduling is a decision making process and is one of the hardcore NP problem. The job-shop scheduling problem (JSSP) consists of ‘n’ jobs and ‘m’ machines. Each job must go through m machines to complete its work. We consider one job consists of m operations. Each operation uses one of ‘m’ machines to complete one job’s work for a fixed time interval. Once one operation is processed on a given machine, it cannot be interrupted before it finishes the job’s work. The sequence of operations of one job should be predefined and maybe different for any job.

Scheduling is allocation of resources efficiently to perform collective tasks. Most of the scheduling problems are combinatorial in nature and solving such problems requires making distinct choices of an optimal solution from number of alternatives. Since the scheduling problems are NP hard, simultaneous optimization of multiple objectives are necessary and it is difficult to find an optimal solution with the use of traditional algorithms. Combinatorial optimization problems are always difficult to solve, due to its high complexity and the computational time increases exponentially.

In the recent years, a lot of research has been carried out on the combinatorial problems and many methods were proposed to overcome the high computational time and the results were being proven that less computational time is required when we use meta heuristics techniques. Meta heuristics has gained popularity in the last decade and it continues to prove that it’s the best technique to get better optimal solutions.

When we hybridize different Meta heuristic approaches the optimal solutions are much more better and effective and efficient with computational time reducing drastically. One such attempt has been made in this paper. An attempt has been made to hybridize three different meta heuristics approach such as Firefly Algorithm (FA), Simulated Annealing (SA), Greedy Heuristics Approach (GHA), thereby developing a new novel technique named as HyFA.

II. LITERATURE REVIEW

There has been a number of research works focused on the Job Shop Scheduling Problem (JSSP) reported in the literature. Over the decades of research, JSSP has undergone extensive research from mathematical approaches to meta heuristic approaches. Table 1 briefly summarizes some of the recent published research works related to JSSP and other recently developed meta heuristics.

Table 1 Brief Literature study on the recent JSSP articles

Authors	Approaches	Problem	Results
Crama et al[1]	Review paper	A review paper that described several combinatorial optimization problems. Serves as a good base paper to understand the basics of Scheduling problems in combinatorial nature	Serves as a good base paper to understand the basics of Scheduling problems in combinatorial nature
Mishra et al [2]	Review paper	A review paper on recent advancements done in non traditional optimization techniques	This paper also serves as a good base paper.
Marichelvam et al [3]	Improved PSO technique + dispatching rule	Addressed the multi-stage hybrid flow shop problem with identical parallel machines at each stage by considering the effect of human factors	A statistical analysis was performed to analyze the results and found their technique is effective
Reddy [4]	Hybrid genetic algorithm	Simultaneous scheduling of machines and AGVs in FMS. Tested on Lawrence instances.	The algorithm presented a good number of diversified and non-dominated solutions
Lin et al [5]	Particle swarm optimization, simulated annealing technique and multi-type individual enhancement scheme	To solve the job-shop scheduling problem for 43 benchmark instances, designed by Fisher and Thompson and instances of LA01–LA40 Lawerence	The proposed algorithm was found to be more robust and efficient than the existing algorithms. The authors also proved that MPSO can reach the optimal area in the search space with smaller population size and iterations than other existing algorithms
Sankar et al [6]	Multi-objective evolutionary algorithm (MOEA) based on GA	Solve the scheduling problem of an FMS with combined objective function of minimizing penalty cost and maximizing machine utilization.	Two versions of the migration model has been constructed and the performance of two parallel GAs has been compared and found it to be effective.
Batur & Erol [7]	A simulated annealing based algorithm has been proposed.	Worked on two-machine flexible robotic cells consisting of CNC machines in which sets of multiple part-types are produced.	Experimental results showed that their approach gave satisfactory results.
Nidish Mathew [8]	NSGA	Benchmark problem of Jerald 43x16 has been attempted to solve multi objectives.	Procedures developed in this work can be suitably modified to any kind of FMS with a large number of components and machines
Kumar et al [9]	Genetic Algorithm and Differential Evolution	Scheduling of FMS systems for 43x16 problem setup.	The results of the proposed approach has been compared with conventional scheduling rules and found to be effective.
Xinyu Li & Gao [10]	Hybrid algorithm (HA) which hybridizes the genetic algorithm (GA) and tabu search (TS)	To minimize the make span. Tested on 201 open problems for FJSP.	The proposed algorithm combines the advantages of evolutionary algorithm and LS method thereby solving the FJSP effectively.
Scaria et al[11]	Artificial Bee Colony (ABC)	Objective to minimize the makespan. Benchmark problems provided by E. Taillard has been solved using this approach	The proposed method found to be more effective than other meta heuristics.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 9 Issue XII Dec 2021- Available at www.ijraset.com

Xiaofeng Li & Zhao [12]	Greedy Algorithm	Flexible flow shop scheduling problem	The authors simplified FFSP into SFFS (Simple FFS) problem for simulation implementation. And found that greedy algorithm to be less complex and effective.
Karthikeyan, Asokan, Nickolas, & Page [13]	Hybrid discrete firefly algorithm (HDFA)	To minimize a combined objective function. Kacem benchmark instances has been used to test the algorithm	Discrete firefly algorithm (DFA) combined with local search (LS) method enhanced the searching accuracy and information sharing among fireflies, thereby making their algorithm more effective.
Mati, Rezg & Xie [14]	Greedy heuristic	Integrated approach with the geometric approach for solving a two-job shop problem, one is flexible job and the second job is a job shop job. Lawrence instanes has been used to test the algorithm	The results obtained by the Greedy heuristic approach on benchmark instances are effective and better.
Sreekara Reddy et al [15]	Multi-objective teacher learning-based optimization algorithm (MOTLBO)	Addresses the multi-objective model for a flexible job shop scheduling problem (FJSSP)	The proposed algorithm has been found to be less complex with less parameter hence effective.
Sun et al [16]	Genetic Algorithm	Problem of penalty function for the job shop scheduling problem. Lawrence instances have been used to test the validity of the proposed approach.	An adaptive penalty function has been designed to search in both feasible and infeasible regions of the solution space. Thereby making the proposed algorithm more effective.
Narendhar, S Amudha, T [17]	Hybrid Bacterial Foraging Algorithm(HBFO)	To minimize makespan for job shop problem. And the algorithm has been tested on Lawrence instances.	The authors compared the HBFO with classic BFO and found their approach gives better make span.
Roshanaei Et al [18]	Artificial Immune and Simulated Annealing (AISA) Algorithms	Position and sequence based mixed integer linear programming (MILP) models have been developed. Case study was used to study the performance of the proposed algorithm.	Parametric size complexities of the proposed models under partial and total cases of F-JSSP were investigated. The proposed AISA found five new best solutions in a benchmark of 20 instances for the make-span optimization criterion.
Yuan and Xu [19]	Hybrid differential evolution (HDE)	To solve flexible job shop scheduling problem (FJSP) to minimize the makespan. Four sets of well known benchmark instances in the FJSP literature are considered.	A novel conversion mechanism has been developed. Also, a local search algorithm based on the critical path is embedded in the DE framework which made the proposed HDE more efficient by finding new best known solutions.

Firefly algorithm (FA) is one of the recently developed swarm intelligence technique which has been proving to be more effective in obtaining optimal solutions. Though Firefly is effective it has a limitation of pre mature convergence. To overcome this problem we have used Simulated Annealing and Greedy Heurustic as local search technique in order to avoid the local optima entrapment. Hence the classic FA is hybridized with SA and GHA and new algorithm Hybrid Firefly Algorithm (HyFA) is proposed. The present work is an attempt to minimize makespan in Job shop scheduling by proposed hybrid technique HyFA . The remainder of this paper is organized as follows. The problem formulation of JSSP with assumpations and constriants are introduced in Section 3. Section 3 also describes the proposed methodolgy to solve the FJSP. Section 4 shows the computational results. Finally, Section 5 provides conclusions and further research.

III. PROBLEM FORMULATION

JSSP are widely known as NP-Hard problem. The JSSP is an operation-sequencing problem on multiple machines subject to some precedence constraints among the operations. The JSSP can be described as a set of n jobs denoted by J_i where i =1,2…n which have to be processed on a set of m machines denoted by M_k where k =1,2….m. Operation of i^{th} job on the k^{th} machine will be denoted by O_{ik} with the processing time P_{ik} . For an n x m JSSP, the problem can be modeled by a set of m machines, denoted by M_k k = (1, 2, ...,m) to process a set of O_{ik} operations.Each job should be processed through the machines in a particular technological order also known as precedence constraint. Once a machine starts to process a job, processing of an operation cannot be interrupted. The required time to complete all operations for their processes is called make-span. Another constraint known, as capacity constraint is that each machine can process only one job at a time. The time span required to complete all operations of all the jobs is known as make span (C_m) and is the most important objective of JSSP.

Minimize Make span C_m = max of [C1, C2, C3,…….Cn] (1)

Where C1, C2, C3 are completion times of Jobs 1,2 and 3

$$O_q \le O_i - t_i \qquad i = 0,1,2,\dots,n \qquad q \in P_i \tag{2}$$

$$\sum_{i \in A(t)} \omega_{im} \le 1 \qquad m \in M, \quad t \ge 0 \tag{3}$$

$$O_i \ge O, \qquad i = 0,1,2,3\dots k \tag{4}$$

The objective fitness function in Eq.(1) is to minimize make span that is the completion time of the last operation. The constraint of precedence relationship is defined by Eq. (2). In Eq.(3), it indicates that one machine can process at most one operation at a time. The finish time must be positive by the constraint stated in Eq. (4).

A. Assumptions

The following assumptions and constraints are to be considered in solving of job shop scheduling problem such as i) all jobs are independent. ii) Machine time includes Job setup time also iii) Job descriptions are known in advance. iv) All machines should be available throughout the process. v) All jobs have to be processed without break. vi) Machine cannot process the parallel job at a time. vii) Each machine will process one job. viii) Each job requires m machines to complete the required process. ix) No Pre-emptions are allowed. The order of processing is not the same and x) Operations cannot be interrupted.

B. HyFA-The hybrid approach implementation

A new hybrid technique HyFA is proposed for solving JSSP with an objective to minimize make-span, which is a combination of Firefly Algorithm(FA), Simulated Annealing (SA) and Greedy Heuristic Approach (GHA). The proposed hybrid HyFA is effective because it's taking advantage of global search and local search as well. We have used two local search methods so that there will be diversity in population when searching for optimal results. The simulated annealing acts as both global and local search method thereby preventing the results to get trapped in local optima. Greedy heuristics resolves the intricacies of sequencing. We therefore expect that the said HyFA will be able give good results for the combinatorial optimization. The flow process for HyFA implementation is exhibited through Figure1.

1) *Firefly Algorithm:* A firefly algorithm is a swarm intelligence optimization technique. A firefly algorithm is applied on the basis of firefly. A firefly is attracted to another firefly based on the brightness of its light [20]. In this algorithm, all the fireflies are assumed to be unisexual. The attractiveness depends upon the brightness, and hence, the node which is less bright is attracted to a node which is brighter and merges together. If there isn't a node nearby, it will move randomly until brightness is detected. It has been used in lots of applications for computational solutions to optimize the process. From the review of the literature it is found that this scheme is used for mostly problem dependent process. In this regards, a new hence, the node that is less bright is attracted to a node that is brighter and merges together. If there isn't a node nearby, it will move randomly until brightness is detected. It has been used in lots of applications for computational solutions to optimize the process. In essence, FA uses the following three idealized rules:[21][22] (1).Fireflies are unisexual so that one firefly will be attracted to other fireflies

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 9 Issue XII Dec 2021- Available at www.ijraset.com

regardless of their sex.(2).The attractiveness is proportional to the brightness and they both decrease as their distance increases. Thus, for any two flashing fireflies, the less brighter one will move toward the more brighter one. If there is no brighter one than a particular firefly, it will move randomly.(3).The brightness of a firefly is determined by the landscape of the objective function. The implementation of FA starts with initialization of swarm of fireflies, with flashing light intensity. At each step, pairwise of light intensity is compared and the firefly with lower light intensity will move toward the higher one. The moving distance depends on the attractiveness. After moving, the new firefly is evaluated and updated for the light intensity. During pairwise comparison loop, the best-so-far solution is iteratively updated. The comparison of pair of fireflies is repeated until termination criteria are satisfied. Finally, the best-so-far solution is visualized. The main processes are described in the following subsections.

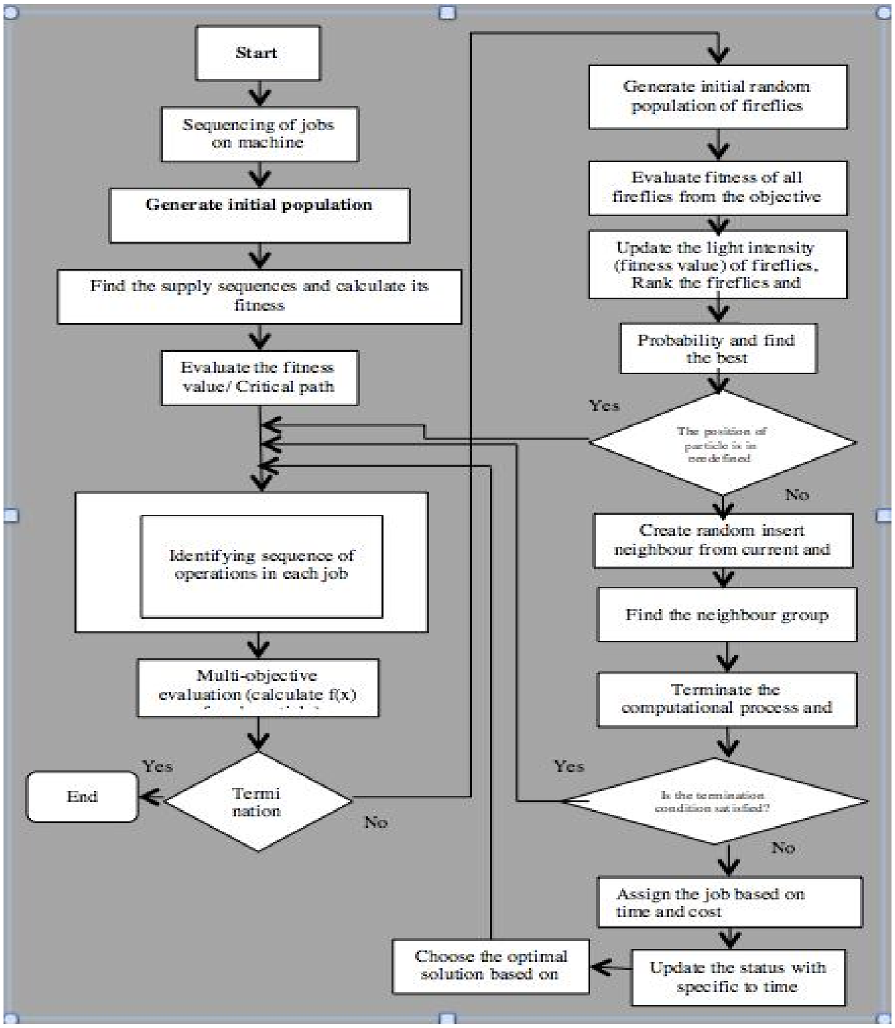


Fig.1. Flow process for HyFA

2) *Solution Procedure*: A random population is introduced initially, for example, every firefly of the underlying populace has been produced utilizing the easy going stage of both the part types to be assigned and the accessible machine on which these part types can be allotted. Alphanumeric strings are used to encode the operations that produce jobs. Each encoded operation is randomly selected and sequenced until all the operations are arranged in order to create a firefly. Here firefly is the candidate solution. So by this random generation we introduce a swarm of fireflies. The length of slots in a firefly is equal to the total number of operations to be performed. The size of the firefly population determines the number of candidate solutions or the amount of search in the solution space. Each firefly is attracted to other firefly based on the objective function in order to achieve optimal solutions. In this paper we are considering minimization problem, so the brightness will be the reciprocal of the Objective Function. The population of the firefly is further evolved when the firefly moves to next attractive firefly (solution).

The distance amongst 2 generic fireflies (e.g. “i" and “j”) is expressed by means of Cartesian distance as follows:

$$r_{ij} = \|pi - pj\| = \sqrt{\sum_{k=1}^{n} (p_{ik} - p_{jk})^2} \quad (5)$$

where pi and pj are spatial co-ordinates.

In the traditional FA the light intensity perceived of each firefly relies on the value of r as stated above and it directly relates to the magnitude of objective function. In line with a formulation provided by Yang (2009), it can be expressed as:

$$I = I_o e^{\gamma r^2} \quad (6)$$

Here, I_o - original light intensity γ - Coefficient of light absorption.

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

$$\beta = \beta_o e^{-\gamma r^2} \quad (7)$$

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

Movement: This factor of the generic firefly “i" that is appealed through a brighter one, can be expressed as:

$$p_i(t+1) = p_i + \beta_o e^{-\gamma r^2 ij}(p_j(t) - p_i(t)^2) + \alpha \epsilon_i \quad (8)$$

In this equation

$p_i(t)$ represents the position of the firefly at the time t;

$\beta_o e^{-\gamma r^2 ij}(p_j(t) - p_i(t)^2)$ 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.

3) *Simulated Annealing:* Simulated annealing is a Meta heuristic technique, which gained popularity due to its ability to be both local search and global search. It was introduced by Kirkpatrick. The algorithm is technically a hill-climbing process except instead of picking the best move, it picks a random move. If the selcted random move provides an improved solution then, the solution will be updated and accepted. Otherwise, the algorithm makes the move but with some probability less than 1. The algorithm also accepts some new solutions by accepting worse neighbours even if it raises the objective. This provides the way to avoid being trapped in local minima and is able to explore global solutions. The success rate of this technique is decided through how effectively and qualitatively the neighbours have been created with this model. From the review of the literature it is found that this scheme is used for mostly problem dependent process. In this regards, a new feasible solution is produced by the random exchange of operation sequence and machine sequence. The process is repeated as many times as possible until the search terminates. The optimal solution is found out in the iteration number below 100.This searching has been done with a maximum of 100 repetitive steps/ iterations. In SA the neighbourhood and annealing rate function and temperature plays an important role to get better performance.
4) *Neighbourhood Structure:* To create a set of feasible solutions, neighbourhood structures are needed. In general, the disjunctive graph is used to describe the neighbourhood structure. In the neighbourhood structures, the neighbourhoods are generated in terms of neighbourhood strategies. As local search efficiency is directly affected by neighborhood structure proper care has to be taken. Random neighbor insert is adopted in our paper. Any two vectors are selected and the larger position number is inserted into previous position of the smaller position number and smaller position number and the subsequent number sequence is postponed. According to the characteristic of the annealing temperature function, it can be obtained that its decreased amplitude becomes faster at a high temperature level, while its decreased amplitude becomes slower at a low temperature level. It is likely to result in an insufficient search in the solution space. In order to enhance the exploitation ability of the SAA, an improved annealing rate function inspired by the Hill function is developed as

$$T(t) = \beta \times \frac{T_0^n}{T_0^n + t^n} \quad (9)$$

5) *Greedy Heuristics:* The greedy heuristics are commonly used to speed up research. This heuristic is one of the popular methods to solve combinatorial optimization and Operations Research problems (i.e. generally NP-hard). The greedy heuristics are used because they are fast, they produce solutions good quality, they are easy to implement and can easily be expanded: in addition, they can be easily randomized [23]. Undeniably, for many cases, this algorithm has proved to lessen the complexity of polynomial time period. In addition, its application has given a lead to avoid the issue called "local optimum". This is the most prevalent method in answering the combinatorial kind of optimizing difficulties and the operational researching issues of NP-hard nature. The particular benefits that are harvested through this approach include the rapidity, solutions with excellence, ease of execution, possibility for expansion, and easier randomization. At the moment which it feels better is picked for selection by this algorithm and hence it derives this name. More precisely, it chooses local optimum point on confidence/ hope and proceeds with an assumption that the chosen point will offer an optimal solution in global manner. Based on the SA and GHA, the mutation, fitness of current state and optimal solution are found. Reiterate the steps until termination criteria are achieved.

The parameters used in HyFA are given in Table 2.

Table 2. Parameters used in HyFA

Parameter	Value
Firefly size in population	30
Annealing function	Fast annealing
Initial temperature and annealing function	100
The total number of generations	200
Initial attractiveness between two fireflies	0.1
Absorption coefficient of the least intensity firefly	1
The maximum iteration size	800

IV. RESULTS AND DISCUSSIONS

This section is devoted to the computational experiments done to evaluate the performance of the proposed HyFA algorithm. The proposed algorithm have been implemented and tested by using Matlab R2016b, computing environment on an Intel Core™i7, with Windows 10. It is aimed to specify with the results that recommended algorithm is an able approach for the scheduling problems involved with flexibility for job shop applications. LA1-25 Benchmark instances were taken from Operations Research (OR) Library to test the efficiency of proposed HyFA with best known solutions (Benchmark datasets) and the obtained datasets are compared with other researcher's experimental research datasets (obtained by using of other algorithms). Our program is editable, we can change number of flies and number of iteration as well, to preset the parameters. The HyFA was run for 20 times inorder to evaluate its consistency and efficacy. Performance comparison of HyFA with other algorithms taken from literature are given in Table 3. Column 1 shows the problem number, column 2 & 3 shows the 'n' jobs 'm' machines , column 4 shows BKS best known solutions, column 5-8 shows the Cm values of other techniques found in literature.

Table 3 Performance Comparison of HyFA with other techniques

Problem	n	m	BKS	GRASP[24]	DE[25]	BFOA[17]	FA[26]	HyFA
LA 1	10	5	666	666	666	693	666	666
LA 2	10	5	655	655	655	683	658	656
LA 3	10	5	597	604	597	634	597	597
LA 4	10	5	590	590	590	624	604	590
LA 5	10	5	593	593	593	593	593	593
LA 6	15	5	926	926	926	926	926	926
LA 7	15	5	890	890	890	903	890	890
LA 8	15	5	863	863	863	873	863	863
LA 9	15	5	951	951	951	951	951	951
LA 10	15	5	958	958	958	958	958	958
LA 11	20	5	1222	1222	1222	NA	1222	1222
LA 12	20	5	1039	1039	1039	NA	1039	1039
LA 13	20	5	1150	1150	1150	NA	1150	1159
LA 14	20	5	1292	1292	1292	NA	1295	1295
LA 15	20	5	1207	1207	1207	NA	1207	1207
LA 16	10	10	945	946	945	NA	945	945
LA 17	10	10	784	784	784	NA	784	784
LA 18	10	10	848	848	848	NA	848	848
LA 19	10	10	842	842	842	894	842	842
LA 20	10	10	902	907	902	NA	902	902
LA 21	15	10	1046	1091	NA	1207	1046	1046
LA 22	15	10	927	960	NA	NA	929	928
LA 23	15	10	1032	1032	NA	NA	1032	1032
LA 24	15	10	935	978	NA	1102	935	935
LA 25	15	10	977	1028	NA	1131	977	977

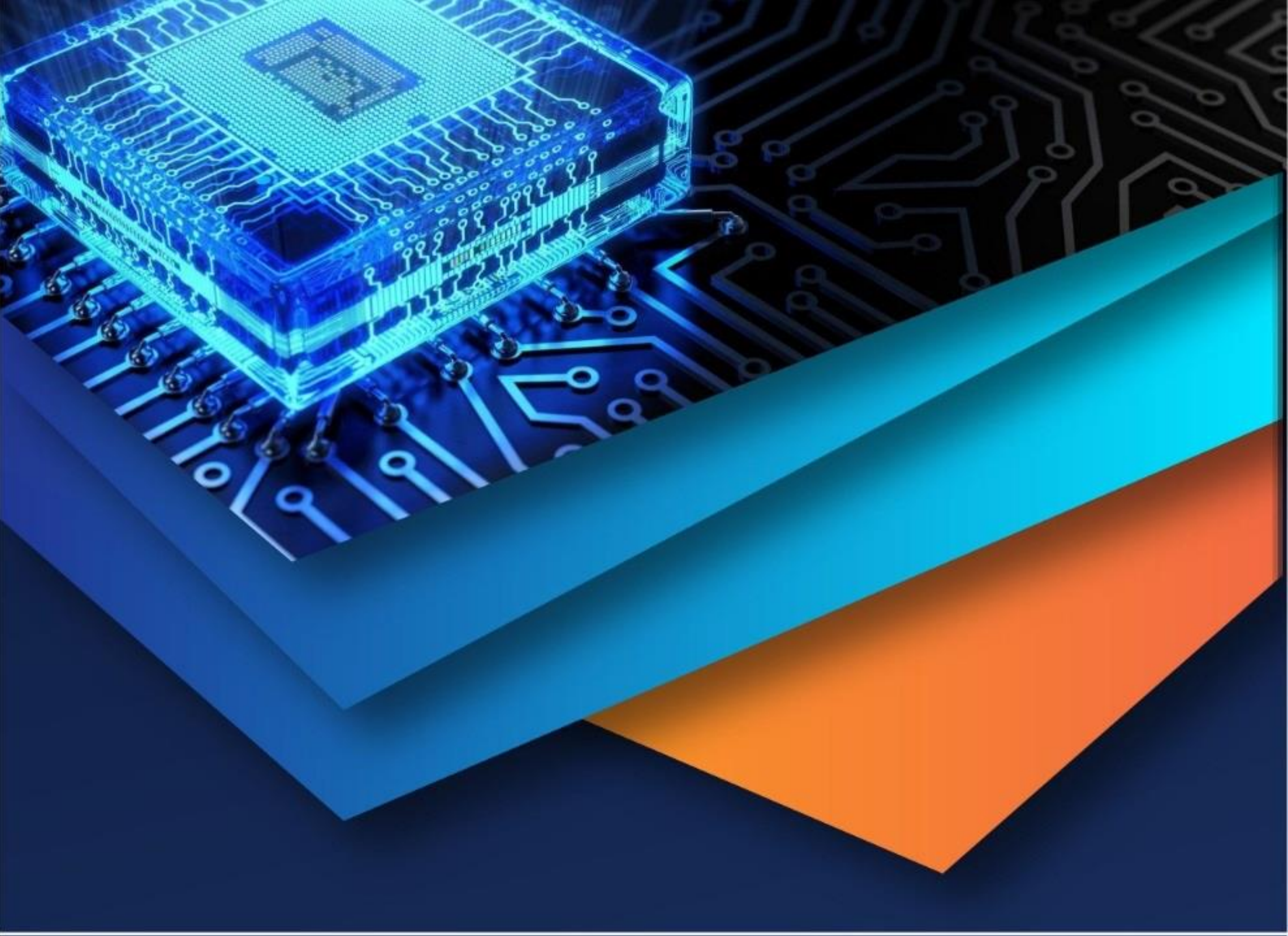
V. CONCLUSIONS AND FUTURE SCOPE

The proposed hybridized algorithm HYFA is used to minimize the make-span of JSSP. The program coding was done in Matlab 2016b and above and the coding can be edited to suit as per the requirement for any combination of machines and jobs. The experimental results have demonstrated a remarkable decrease in computing time and getting better optimal operation and machine sequence for solving the JSSP. The contributions of this research paper are:

1) A novel hybridized evolutionary algorithm HyFA is done by combining three Meta heuristics namely FA, SA, GHA.
2) The proposed algorithm showed an effective result and successfully solved the scheduling problem. Twenty-Five test instances of Lawrence with different size are taken to conduct experimental studies. The objective of minimizing make span have been attempted and the optimal results were found in very less computational time.
3) In the future, attempts can be made further, to hybridize other evolutionary algorithms to construct an effective hybrid algorithm. Also we can consider other objectives and multi-objectives and try to reduce the constraints and solve the other branches of JSSP like FJSP. Not all benchmark problems have optimal solutions. We hope this technique will help to achieve that.

BIBLIOGRAPHY

[1] Crama Y (1997) Combinatorial optimization models for production scheduling in automated manufacturing systems. Eur J Oper Res 99:136–153. https://doi.org/10.1016/S0377-2217(96)00388-8
[2] Mishra R, Devi GK, Madan A (2019) Non-traditional optimization techniques of scheduling in FMS: A Review. Int J Res Eng Innov 3:157–164
[3] Marichelvam MK, Geetha M, Tosun Ö (2020) An improved particle swarm optimization algorithm to solve hybrid flowshop scheduling problems with the effect of human factors – A case study. Comput Oper Res 114:104812. https://doi.org/10.1016/j.cor.2019.104812
[4] Reddy BSP, Rao CSP (2006) A hybrid multi-objective GA for simultaneous scheduling of machines and AGVs in FMS. Int J Adv Manuf Technol 31:602–613. https://doi.org/10.1007/s00170-005-0223-6
[5] Lin TL, Horng SJ, Kao TW, et al (2010) An efficient job-shop scheduling algorithm based on particle swarm optimization. Expert Syst Appl 37:2629–2636. https://doi.org/10.1016/j.eswa.2009.08.015
[6] Sankar SS, Ponnanbalam SG, Rajendran C (2003) A multiobjective genetic algorithm for scheduling a flexible manufacturing system. Int J Adv Manuf Technol 22:229–236. https://doi.org/10.1007/s00170-002-1464-2
[7] Batur GD, Erol S (2016) Using simulated annealing for flexible robotic cell scheduling. Gazi Univ J Sci 29:573–582
[8] Nidhish Mathew Nidhiry DRS (2012) A COMBINED OBJECTIVE GENETIC ALGORITHM FOR SCHEDULING OF MACHINES IN FMS. Int J Eng Res Appl ISSN 3:
[9] 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
[10] Li X, Gao L (2016) An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem. Int J Prod Econ 174:93–110. https://doi.org/10.1016/j.ijpe.2016.01.016
[11] Scaria A, George K, Sebastian J (2016) An Artificial Bee Colony Approach for Multi-objective Job Shop Scheduling. Procedia Technol 25:1030–1037. https://doi.org/10.1016/j.protcy.2016.08.203
[12] Li X, Zhao H (2009) Greedy Algorithm Solution of Flexible Flow Shop Scheduling Problem. Int J Comput Sci Netw Secur 9:9–12
[13] Karthikeyan S, Asokan P, Nickolas S, Page T (2015) A hybrid discrete firefly algorithm for solving multi-objective flexible job shop scheduling problems. Int J Bio-Inspired Comput 7:386. https://doi.org/10.1504/IJBIC.2015.073165
[14] Mati Y, Rezg N, Xie X (2001) An integrated greedy heuristic for a flexible job shop scheduling problem. Proc IEEE Int Conf Syst Man Cybern 4:2534–2539. https://doi.org/10.1109/ICSMC.2001.972939
[15] Sreekara Reddy MBS, Ratnam C, Rajyalakshmi G, Manupati VK (2018) An effective hybrid multi objective evolutionary algorithm for solving real time event in flexible job shop scheduling problem. Meas J Int Meas Confed 114:78–90. https://doi.org/10.1016/j.measurement.2017.09.022
[16] Sun L, Liang YY, Cheng X (2010) Solving Job Shop Scheduling Problem Using Genetic Algorithm with Penalty Function. Int J Intell Inf Process 1:65–77. https://doi.org/10.4156/ijiip.vol1.
[17] Narendhar S, Amudha T (2012) A Hybrid Bacterial Foraging Algorithm For Solving Job Shop Scheduling Problems. Int J Program Lang Appl 2:1–11
[18] Roshanaei V, Azab A, Elmaraghy H (2013) Mathematical modelling and a meta-heuristic for flexible job shop scheduling. Int J Prod Res 51:6247–6274. https://doi.org/10.1080/00207543.2013.827806
[19] Yuan Y, Xu H (2013) Flexible job shop scheduling using hybrid differential evolution algorithms. Comput Ind Eng 65:246–260. https://doi.org/10.1016/j.cie.2013.02.022
[20] Lindfield G, Penny J (2017) The Firefly Algorithm. In: Introduction to Nature-Inspired Optimization. Elsevier, pp 85–100
[21] Fister I, Yang XS, Brest J (2013) A comprehensive review of firefly algorithms. Swarm Evol Comput 13:34–46. https://doi.org/10.1016/j.swevo.2013.06.001
[22] Yang XS, Karamanoglu M (2013) Swarm Intelligence and Bio-Inspired Computation: An Overview. In: Swarm Intelligence and Bio-Inspired Computation. Elsevier, pp 3–23
[23] El-Ghazali Talbi (2009) Metaheuristics. University of Lille
[24] Binato S, Hery WJ, Loewenstern DM, Resende MGC (2002) A grasp for job shop scheduling. Oper Res Comput Sci Interfaces Ser 15:59–79. https://doi.org/10.1007/978-1-4615-1507-4
[25] Shilpa P, Balaraju G (2018) Job shop scheduling using differential evolution algorithm. Int J Mech Prod Eng Res Dev 8:327–338. https://doi.org/10.24247/ijmperdjun201837
[26] Udaiyakumar KC, Chandrasekaran M (2014) Application of firefly algorithm in job shop scheduling problem for minimization of Makespan. Procedia Eng 97:1798–1807. https://doi.org/10.1016/j.proeng.2014.12.333



10.22214/IJRASET

INDEX COPERNICUS
45.98

IMPACT FACTOR:
7.129

TOGETHER WE REACH THE GOAL
IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY