SCHEDULING IN FLEXIBLE MANUFACTURING SYSTEM (FMS) USING SOFT COMPUTING TECHNIQUES

A THESIS
SUBMITTED IN FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

DOCTOR OF PHILOSOPHY
IN
MECHANICAL ENGINEERING

By

GAYATHRI DEVI K
(2K16/Ph.D/ME/61)

UNDER THE SUPERVISION OF
Prof. R. S. MISHRA

DEPARTMENT OF MECHANICAL ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
NEW DELHI-110042, INDIA

MARCH 2023

DELHI TECHNOLOGICAL UNIVERSITY
Bawana Road, Delhi- 110042

CANDIDATE'S DECLARATION

I hereby declare that the research work presented in this thesis, entitled **"Scheduling in Flexible Manufacturing System (FMS) Using Soft Computing Techniques"** is an original work carried out by me under the supervision of Prof. R.S. Mishra., Professor, Department of Mechanical Engineering, Delhi Technological University, Delhi. This thesis has been prepared in conformity with the rules and regulations of the Delhi Technological University, Delhi. The research work reported and results presented in the thesis have not been submitted either in part or full to any other university or institute for the award of any other degree or diploma.

Place: Delhi
Date:

Gayathri Devi K
(2K16/Ph.D/ME/61)
Research Scholar
Department of Mechanical Engineering
Delhi Technological University, Delhi

DEPARTMENT OF MECHANICAL ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

Bawana Road, Delhi- 110042

CERTIFICATE

This is to certify that the work embodied in this thesis entitled, **“Scheduling in Flexible Manufacturing System (FMS) Using Soft Computing Techniques”** being submitted by **Gayathri Devi K (Roll No- 2K16/PhD/ME/61)** for the award of **Doctor of Philosophy Degree (Ph.D.) in Mechanical Engineering** at Delhi Technological University, Delhi is an authentic work carried out by her under our guidance and supervision.

It is further certified that the work is based on original research and the matter embodied in this thesis has not been submitted to any other university/institute for award of any degree to the best of our knowledge and belief.

Place: Delhi

Date:

Prof R.S. Mishra
SUPERVISOR
Professor, Dept. of Mechanical Engineering (DTU, Delhi)

ACKNOWLEDGEMENT

This thesis has been written during my position as a doctoral candidate at Delhi Technological University, Delhi. Foremost, I am thankful to the **Almighty**, most beneficial and merciful, who blessed me with strength and patience to acquire knowledge and learning.

I would like to express my sincere gratitude to my supervisor, **Prof R.S. Mishra** for his continuous support of my Ph.D. study and research. Prof Mishra guided me with his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis and I am honored to acknowledge mentoring along the way. I thank him for the scholarly inputs which I received throughout the research work. I sincerely treasure his precious remarks about academic writing, research, and future perspectives.

I would like to express my sincere gratitude to the members of the dissertation committee. I am extremely grateful for the members willingness to take part in my doctoral examination, as well as for the precious comments and insightful questions they provide on the present research work. I also wish to express my gratefulness to "The Department of Mechanical Engineering, Delhi Technological University, Delhi" for facilitating the progress of my research work by providing the necessary facilities in the department.

Ph.D. is not a lonely task. I would like to express my deep gratitude to **Prof. J.Jerald** from NIT, Trichy whose insights and expertise have been instrumental in shaping my research and making it more meaningful and **Prof Nandakumar. C** from MIT, Chennai for his unwavering support and valuable guidance during the challenging parts of my research.

I would also like to appreciate my Ph.D. colleagues **Mrs. Gagan Preet Kaur, Mr. Mohd Rizwan, Dr. Roshan Raman, Mr. Sumit Jain** for their priceless moral support and to put up with my constant ramblings about my research.

Lastly, I am truly blessed to have such amazing parents, **Mr. Karunagaran and Mrs. Shanthi,** who have always encouraged me to pursue my dreams and never wavered in their support. I thank them for their countless sacrifice, motivation and love

for me. To my husband **Rajkumar**, I am forever grateful for your unwavering support and encouragement. You have been my rock throughout this journey, offering words of encouragement when I needed it most, and always believing in me even when I didn't believe in myself. Thank you for being my partner in every sense of the word and for being there for me every step of the way.

To my amazing children **Rahul and Raagav**, thank you for being my motivation and inspiration. Your love, laughter, and joy have been the light in my life and have kept me going during the long hours of research and writing. Your love, support, and patience have been a constant source of strength for me, and I am grateful for your understanding during those times when I couldn't be fully present. I hope that this accomplishment will inspire you to pursue your own dreams with the same passion and dedication. I love you more than words could ever express.

ABSTRACT

In today's highly competitive and fast-paced manufacturing industry, companies are increasingly turning to flexible manufacturing systems (FMS) to improve their efficiency and productivity. FMS is an automated manufacturing system that includes transport vehicles, automated storage, and a comprehensive computer control system, all working together to produce a wide variety of parts quickly and efficiently. FMS is a critical component of Industry 4.0, the fourth industrial revolution characterized by the integration of advanced technologies making it smart manufacturing process.

Scheduling optimization is a crucial aspect of Flexible Manufacturing Systems (FMS) that involves determining the optimal sequence for producing multiple components and allocating the appropriate resources to each operation. The FMS scheduling optimization is of paramount importance for manufacturers, as it results in increased productivity and reduced production costs. By utilizing an efficient FMS scheduling optimization, manufacturers can achieve faster production times, higher throughput rates, and improved quality control. The optimization of FMS scheduling is a significant factor in the current Industry 4.0. The integration of advanced technologies with FMS scheduling optimization can lead to the development of smarter factories with improved efficiency, accuracy, and automation. As such, the optimization of FMS scheduling is a vital element in the success of modern manufacturing operations.

Traditional optimization methods, such as linear programming and dynamic programming, have been used for scheduling optimization in manufacturing for several decades. However, these methods have limitations when it comes to solving complex scheduling problems in Flexible Manufacturing Systems (FMS), which are characterized by large

search spaces, non-linear relationships, and combinatorial constraints. Metaheuristics, a class of optimization algorithms that use heuristic rules to explore the search space efficiently, have emerged as a powerful tool for solving complex FMS scheduling problems. Metaheuristic algorithms are inspired from natural phenomena and mimics it to find near-optimal solutions by iteratively exploring the search space, making probabilistic moves, and adapting to the search environment. These algorithms can handle multiple objectives, constraints, and uncertainty, making them suitable for FMS scheduling optimization. With the advancement of computing power and the availability of high-performance computing platforms, metaheuristic algorithms have become even more useful in FMS scheduling optimization.

In this research, three novel hybrid meta heuristic methods have been proposed: 1) GAPSOTS- *An amalgamation of Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Tabu Search (TS)* 2) HAdFA- *The Hybrid Adaptive Firefly Algorithm* and 3) HFPA- *Hybrid Flower Pollination Algorithm*. GAPSOTS is simple hybridization of classic meta heuristics without any adaptive features. The GAPSOTS suffered from local optima entrapment and convergence was impetuous. To address the premature convergence problem inherent in the classic Firefly Algorithm (FA), the researcher developed HAdFA that employs two novel adaptive strategies: employing an adaptive randomization parameter (α), which dynamically modifies at each step, and Gray relational analysis updates firefly at each step, thereby maintaining a balance between diversification and intensification. HFPA is inspired by the pollination strategy of flowers. Additionally, both HAdFA and HFPA are incorporated with a local search technique of enhanced simulated annealing to accelerate the algorithm and prevent local optima entrapment.

The current study addresses FMS scheduling optimization for the following:

- A Flexible Job Shop Scheduling Problem (FJSSP) was analysed, studied and tested with proposed meta-heuristics for several benchmark problems for multi-objectives of makespan (MS_{max}), maximal machine workload (WL_{max}), total workload (WL_{total}), total idle time (T_{idle}) and Total tardiness, i.e., lateness of jobs (T_{late}).
- An FMS configuration, integrated with AGVs, Automatic storage and retrieval system (AS/RS) has been optimized using a Combined Objective Function (COF) with the aim of minimizing the machine idle time and the total penalty cost combinedly. In order to test the effectiveness of this optimization method, several problems were developed and tested by varying the number of jobs and machines for this particular FMS setup.
- The concurrent scheduling of machines and AGVs in a multi-machine FMS setup for different layouts has been studied. This problem has been developed as a multi-objective optimization with objectives to minimize the makespan, mean flow time, and mean machine idle time. Proposed meta-heuristics have been employed and tested on randomly generated example problems to evaluate their performance for this setup. These meta-heuristics have proven to be effective in finding optimal solutions, and their application can lead to improved efficiency and reduced costs in FMS setups.
- Finally, a real-life case study was conducted in a Lube Oil Blending Plant, Faridabad, India. The proposed GAPSOTS and HAdFA are tested for three problems with varying jobs and machines for multi objectives.

The corresponding computational experiments have been reported and analyzed. The suggested algorithms have been implemented and tested using Matlab R2019a, computing environment on an Intel Core™i7, with Windows 10. The results indicate that the proposed HAdFA tends to be more efficient among the proposed algorithms and consistently demonstrated to achieve not only optimal solutions but also new makespan values were found for some problems. The efficiency of HAdFA can be attributed to the adaptive parameters integrated into it. This algorithm significantly improves convergence speed and enables the exploration of a large number of rich optimal solutions.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

FMS	Flexible Manufacturing Systems
CNC	Computer Numerical Controlled
CMM	Coordinate Measuring Machines
AGV	Automated Guided Vehicle
SMC	Single Machine Cell
FMC	Flexible Manufacturing Cell
GA	Genetic Algorithm
PSO	Particle Swarm Optimization
TS	Tabu Search
GAPSOTS	Hybridization of GA, PSO, TS
FA	Firefly Algorithm
HAdFA	Hybrid Adaptive Firefly Algorithm
FPA	Flower Pollination Algorithm
HFPA	Hybrid Flower Pollination Algorithm
AHP	Analytic Hierarchy Process
SAW	Simple Additive Weighting
WPM	Weighted Product Method
CPN	Coloured Petri Net
NV	Nearest Vehicle
MAS	Multi-Agent Systems
CS	Cuckoo Searching
PSO	Particle Swarm Optimization
ORCA	Optimized And Reactive Control
PN	Petri Net
TMG	Timed Marked Graph
LP	Linear Programming
FAMS	Flexible Automotive Manufacturing System
AI	Artificial Intelligence
CAPP	Computer Aided Process Planning
ET	Evolutionary Techniques

TS	Tabu Search
DTP	Double Tabu Search
BFOA	Bacterial Foraging Optimization Algorithm
DE	Differential Evolution
AIS	Artificial Immune Systems
GA	Genetic Algorithm
MOS	Multiple Objectives Symbiotic
MFPP	Multi-Objective FMS Process Planning
MOSEA	Multi-Objective Symbiotic Evolutionary Algorithm
NSGA	Non-Dominated Sorting Genetic Algorithm
RDWS	Robust Dynamic Window Search Algorithm
PSM	Particle Swarm Method
SA	Simulated Annealing
SVM	Support Vector Machine
ACO	A Novel Ant Colony Optimization
VNS	Variable Neighborhood Search
SAA	Simulated Annealing Algorithm
DR	Dispatching Rules
AdFA	Adaptive Firefly Algorithm
FJSSP	Flexible job shop scheduling problem
COF	Combined Objective Function
DE	Differential Evolution
MOPSO	multi- objective particle swarm optimization
BEG NSGA II	bee evolutionary guiding nondominated sorting genetic algorithm II
RRHC	Random Restart Hill Climbing
ADCSO	Adaptive discrete cat swarm optimization algorithm
IH-PSO	Improved hybrid particle swarm optimization algorithm
DP DATA	Dauzère–Pérès data set
hGA	Hybrid Genetic Algorithm
HTSA	Hybrid tabu search algorithm
MODE	Multi objective Differential Evolution
DFA	Discrete Firefly Algorithm

CHAPTER 1

INTRODUCTION

1.1 Background

Till the period of 20^{th} century, the manufacturing industries did not offer any flexibility. In fact, there was no demand in terms of efficiency because the competitions were only among the national markets and not the international markets. Manufacturers had decided the things that consumers buy. Literally, there were no choices or options for the customer and only the manufacturers influence the buying capacity or desire of any consumer. The World War II had paved way to new-flanged materials and led to innovative technologies in production in terms of output and quality. It had opened the floor to overseas market and competition. It was that time when the paradigm shift had occurred in the open market from the manufacturer to the customer.

In the year 1965, Theo Williamson had received the patent for the very first Flexible Manufacturing Systems (FMS) for his invention of numeric control equipment. CNC machines utilized for lathe and mill applications are few examples that are covered under the category of numeric control machines. The huge development and changes in the technologies equally had imposed some challenges and difficulties in the period around 1970 and obviously the flexible manufacturing had gained popularity and manufacturers started using it for adapting to the new environment/ changes. It was in the year 1980, the manufacturers, first time in the history, had focused the quality, efficacy and flexibility to sustain in the market.

1.2 FLEXIBLE MANUFACTURING SYSTEM

The increased demand for the manufacturing needs and to satisfy the huge consumers of multiple domains, effective manufacturing is essential, which incorporates Flexible Manufacturing System (FMS). Production or productivity is paramount in any manufacturing industry. The resources that are utilized for production in a manufacturing unit may involve unprocessed materials, capital investment, workforce / human resource, logistics / transportation of goods, etc.

1.2.1 Scheduling in Manufacturing System

Schedule is a document and normally states the occurrence of things and illustrates a strategy for the timing of definite actions and Scheduling is the method of creating the schedule. In general, the problem of scheduling can be handled in two stages; in the first stage, the order is planned or decided to select the subsequent work. In the second stage, formation of starting time and possibly the finishing time of every job is made.

1.2.2 Production Scheduling

Effective production is ensured through a process called "production scheduling". It is defined as an activity in which the resources are assigned with a proper timescale or timetable and the time of operation of each and every associated activity is scheduled or organized. It finds the type of resource to be expended at a particular manufacturing phase and as per the evaluations a time-table is made because of which the organization will not have any resource deficit during the production time. The FMS is a scheme of manufacturing that permits several products produced without any reconfiguration requirements of the entire manufacturing line. It possesses a group of numeric controlled machines having multiple function facilities. Further it contains a material management scheme and the computer system connected online through which, the governing and controlling of the complete system becomes simple and easy [1].

Production scheduling is usually thought as the utmost important concern in the planning and functioning of a manufacturing scheme. Enhanced scheduling scheme shows greater effect on cost saving, enlarged productivity, gratified customers, etc. Further, new consumer demand for high range goods has led to a growth in product complication, which additionally stresses the necessity for better-quality scheduling. Expert scheduling increases the volume consumption efficiency and decreases the time needed to complete tasks and it ultimately helps the organization in huge profit making especially in the prevailing competitive atmosphere [2].

The manufacturing industries must improve the production and manufacturing schemes to satisfy the variable demands of the clients for custom-built products and to withstand the present day market competitions worldwide. The scheme should be adaptable, productive and capable of serving the demands within stipulated/ committed time at a reasonable price. The FMS integrates the benefits of job-shop and batch manufacturing schemes. There is an option for changing the parts and tools in an automated manner by means on controllable and centralized computer in case of FMS [3].

FMS involves certain amount of flexibility and permits the system to respond in the event of expected and unexpected changes. Flexibility is the rate at which it responds to the change and adapts it. To be treated as flexible, the flexibility should happen in the product's lifespan as a whole. That is, starting through the design to manufacturing to delivery. It can be reiterated that FMS is a computer regulated scheme, which can generate a range of products or else parts in any sequence without taking much of time to change the order of sequence.

1.2.3 Types of Flexibilities

The FMS design should incorporate concurrent manufacturing of several capacities of a changing variety of products/ goods with superior quality. Possibly there are 11 types of flexibilities, which are listed below [4]:

Machine Flexibility: It refers to the flexibility offered by the machine with various operations that could be performed.

Product Flexibility: It is the capability to include new products/goods in the scheme.

Router Flexibility: It is the capability to use various routes for yielding a product.

Material Handling Flexibility: It is the capacity of transferring the products in a given facility of manufacturing.

Operation Flexibility: It is the capacity to create a product in multiple methods.

Process Flexibility: It is the capacity of the system for creating a group of products.

Volume Flexibility: It is the ability of the system to make low and high amount of products within the economic viability for a predetermined investment.

Expansion Flexibility: It is the adaptability of the system to enlarge/ expand for increasing the amount of production.

Program Flexibility: It is the capacity of running the system in an automated manner.

1.2.4 Advantages of FMS

The various benefits of FMS can be listed as under [5] :

1. Realizing an extremely automatic manufacturing process with complete monitoring by the use of computers

2. Quality management and increased productivity

3. Possibility for scaling the operations for various output stages

4. Economical way of reconfiguration and customization of the procedure of manufacturing

5. Timely report of the process of manufacturing with thorough data

6. Coordination amongst the manufacturers, suppliers, and consumers to streamline the work, reduce the cost and to improve the efficiency

1.2.5 Disadvantages of FMS

FMS possesses few drawbacks, which can be listed as [6],

1. Complexity in implementation

2. Requires experts for operating the machineries and for maintenance

3. Requires huge capital investment

1.3 MAJOR COMPONENTS OF FMS

The FMS contains the following important elements [7]:

Workstation: It comprises of Computer Numerical Controlled (CNC) machines to carry out several functions on set of parts. It even contains additional work stations such as, checking stations, assembly mechanisms and sheet metal presses.

Automated Material Handling and Storage system: The material handling system operates in an automatic manner and is useful to move the work and subassembly parts amongst the stations for processing. It includes the handling devices that works automatically such as, automated guided automobile, conveyors, etc. Actually, there are two categories of systems for material handling namely, the primary and secondary systems.

Primary handling system: It is accountable to transfer the work parts among the system stations and establishes the fundamental arrangement of FMS.

Secondary handling system: It covers various devices available at the FMS workstations, which may include the transferring/ moving instruments, automated pallet changer and related devices.

Computer Control System: It is helpful in controlling the actions of the handling stations and the material treatment arrangement in FMS.

Inspection Equipment: It takes account of the Coordinate Measuring Machines (CMMs) utilized for the purpose of inspection when not connected online and is computerized

Miscellaneous Components

It comprises of a centralized coolant and proficient chip separating scheme and possesses the following features:

1. The system should have the potential to regain the coolant
2. The mixture of parts, pallets, and the fixtures are to be dressed then and there by removing the dust, chippings prior to the working and before inspection.

1.4 FUNCTIONS OF SCHEDULING

The functions that are to be performed methodologically and in a well-organized way for achieving appropriate and effectual design of production scheduling are [8]:

1. Allocating various works to several facilities by viewing the viability of allocation (loading).

2. Creating a rule set with regard to priorities to order or arrange the activities/ actions on the facilities (sequencing)

3. Send out work orders according to the schedule to induct loading of works to facilities

1.5 GENERAL OBJECTIVES OF SCHEDULING

Following are the goals of scheduling in FMS [9]:

1. Realizing maximum efficiency of the operations by means of optimal use of machineries and apparatus. (Performance based)
2. Retaining fewer inventories in raw materials as well as in the process. (Material based)
3. Sustaining small flow-time of products/ goods. Identify potential bottlenecks in their production processes and take corrective actions (due-date based)

The objectives are frequently contradictory to each other and by itself the scheduling procedure should have a compromise between said objectives in such a way that appropriate equilibrium is attained.

1.6 SCHEDULING ELEMENTS

The sequence and timing of operations are decided through scheduling through which, the usage of resources is made optimum and the production needs are met. The scheduling elements are listed below.

a. Configuration of influx jobs
b. Type and quantity of machineries and operations
c. The worker/machine or worker/operation ratio
d. Job flow design
e. The priority rules for assigning the work.

1.7 METHODS TO SOLVE FMS SCHEDULING

Usually, the goal of FMS scheduling does not only consider one factor rather it involves various objectives. In simple terms, the problem of FMS scheduling is modelled as a multi-objective problem. This necessitates the formation of mathematical equations with various restrictions/ constraints. Solving those problems involving simple mathematics is very cumbersome and time consuming. Hence, various other methods are suggested to solve the same in the literatures. Few of them are listed under [10]–[13].

1. Mathematical programming procedure
2. Multi criteria decision making
3. Heuristic methods
4. Control theoretical model
5. Simulation based model
6. Artificial intelligent based technique
7. Meta-heuristic scheme

1.7.1 Mathematical Programming Procedure

This technique involves, linear programming, branch algorithm, bound algorithm, etc. This kind of model works well when the problem size is small and is not genuine for large sized problems. Also, the model is constructed using initial guesses and approximations, which are not validated in real-life problems.

1.7.2 Multi Criteria Decision Making

The several factors linked to FMS are devised as goals in this type.

Goal programming and Integer programming formulations are few examples. The classic goals, which are frequently considered, will fulfil the production requests, dipping the time of output of parts and harmonizing the machine use.

1.7.3 Heuristic Method

The heuristics framed are in the form dispatching rules, sometimes a combination of such rules are used increasing the complexity. The rules to be combined are decided by the alternative routes available. The procedures involved with this type are iterative in most of the cases and is used to estimate the best route out of the probable routes available.

1.7.4 Control Theoretical Model

The main theme of this model is to sustain and safe guard the buffer of parts formed in FMS till the feasibility limit. This approach locates the solution in the FMS's production capacity boundary. Together with the security buffer level for all part kinds, a corresponding state for capacity is created for each machine state.

1.7.5 Simulation Based Model

Ingalls [14] has quoted the definition of simulation stated by Shanon as "the process of designing a model of a real system and conducting experiments with this model for the purpose either of understanding the behaviour of the system or of evaluating various strategies (within the limits imposed by a criterion or set of criteria) for the operation of the system".

In this methodology, simulation is suggested as a tool to estimate various rules linked to dispatching. A simulation model is prepared to signify the manufacturing arrangement and the model's rules are verified then. This method is considered powerful as it offers a feasible answer.

1.7.6 Artificial Intelligent Based Technique

This is used to unravel complex real-time optimization issues, which possess a big searching space. These Expert systems are applied to imitate the performance of a skilled human. Definite systems are regularly made for well-defined manufacturing systems. These methods have provided worthy outcomes for area specific issues.

1.7.7 Meta-heuristic scheme

A Meta-heuristic is an all-purpose solution technique that delivers an overall structure and policy strategies for evolving heuristic method for fitting a specific problem. They syndicate heuristics in a more common structure to direct the searching practice in capably exploring the searching space to get a nearer ideal solution. Therefore, these approaches are found appropriate for solving even problems of larger size with less effort and time.

The common optimization techniques can be classified as shown in Figure 1.1 and taxonomy of meta-heuristic approaches is depicted in Figure 1.2.

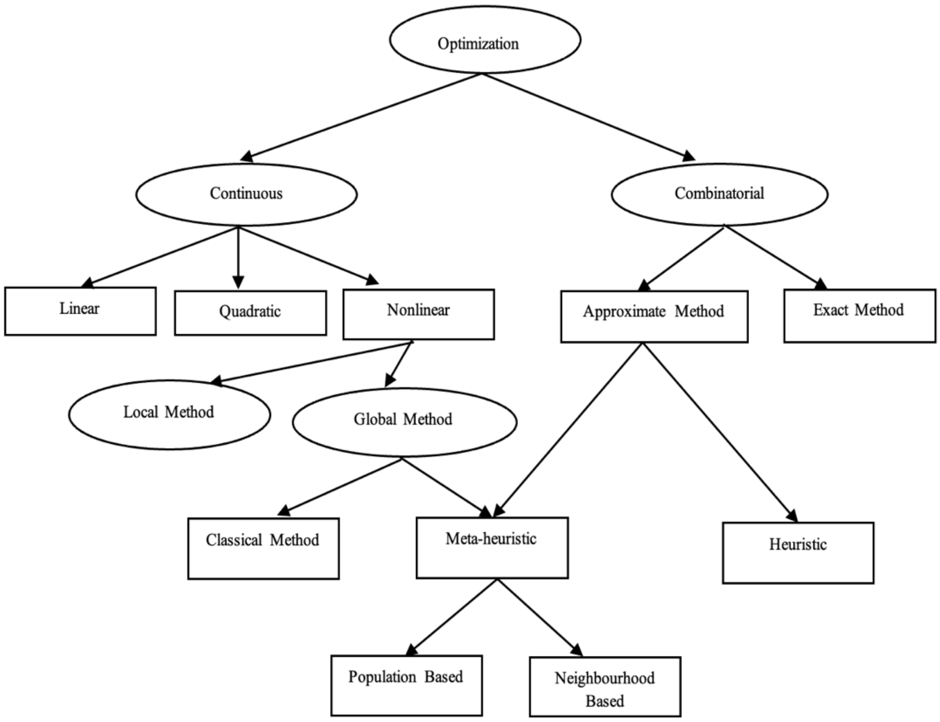


Figure 1.1 Classification of common search practices

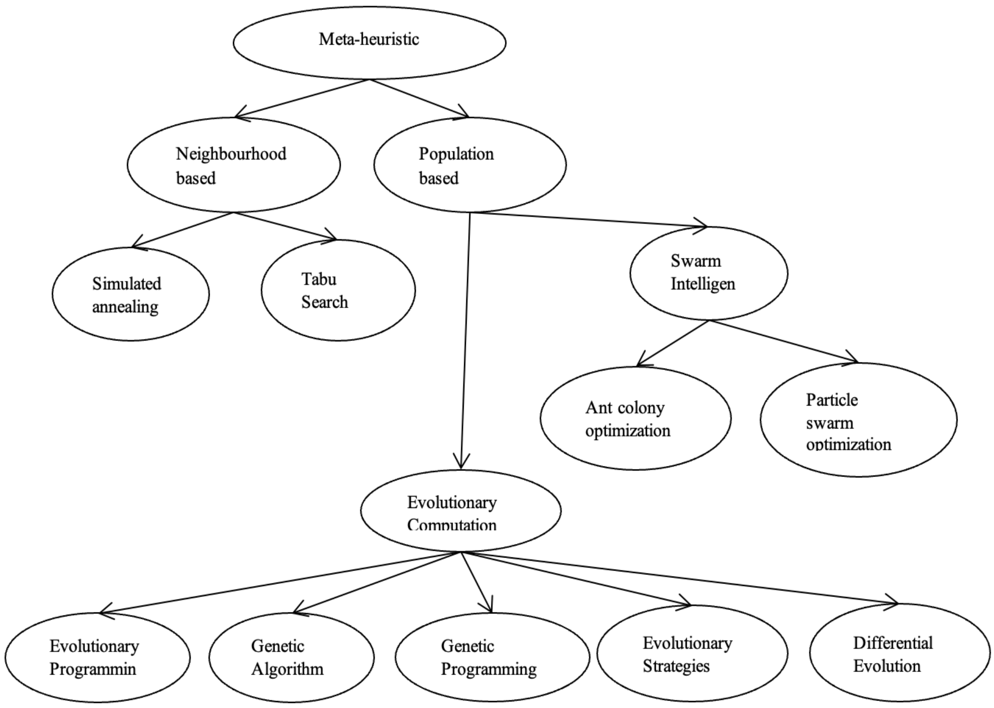


Figure 1.2 Classification of Meta-Heuristic Approach

1.8 PERFORMANCE MEASURES

For solving the problem associated with scheduling of FMS, proper objectives are required and the problem is modelled as a multiple objective problem. There may be numerous objectives, which are complex in nature and sometimes the objectives may be contradictory. Hence for stating the objectives clearly, the performance measure of a particular scheduling is essential. A regular performance measure is one, which does not have a decreasing function in terms of job completing period. For such measures, the objective function is usually the minimizing of performance measure. Some of the important performance measures include, total flow time, makespan (overall length of the schedule), overall delay, quantity of delayed jobs and maximum delay, etc. The makespan factor is very vital in scheduling as it relates to the productivity improvement and highest resource usage. In current scenario, every customer is satisfied only when a product is delivered quickly or when it is delivered within the stipulated date/ time. Thus, on-time delivery becomes a significant factor in terms of gratifying the customers. Unless or otherwise the said factor is not focused it would result in losing the customers and market attraction. Also, the production cost is obviously important for any company. Production cost lessening is realized with proper utilization of resources (material management), less time to complete the production and delivering the products on-time. Another kind of performance measure is the non-regular one. This is something related to job earliness. When the jobs are completed well before the time stated then the jobs are penalized.

Thus, a problem may contain purely regular performance measures or else it is may be combinational. Most of the problems associated to scheduling in real time are multi-objective in general. The earlier and few present-day researches had been focusing with shop scheduling problem with only one objective of reducing the makespan [15], [16]. The analysis will not yield genuine results if the scheduling is modelled with one objective in the present scenario. Present day's manufacturing is highly dynamic and has an inconsistent atmosphere. Thus, the problem of scheduling should be modelled as a multi-objective problem. In most of the recent literatures, even though multiple objectives are considered, it just involves two or three criteria [17]–[19]. For rigorous analysis and optimal design many objectives should be included. Once the objectives are made by including proper number of valid performance measures, a hybrid meta-heuristic algorithm is used for optimization or meet the objective in an efficient manner.

1.9 STATEMENT OF THE PROBLEM

Flexible manufacturing systems (FMSs) have developed as an exceedingly effectual manufacturing scheme to create products of different sizes and range [20]–[22]. In a common perspective, flexibility can be regarded as a character of the linking among a system and its periphery. FMS scheduling contains the elements/ components of machine-loading, routing of parts, scheduling the tasks related to manufacturing, planning and assigning the tools, forming the agenda for buffer utilization and scheduling the automatic guided vehicles (AGV) [23]. These literatures have not given importance to the setup time/ cost or part process-time. Neglecting these factors simplifies the analysis, which may not affect the solution quality for few applications. Yet, most of the application should include these features as the solution is highly dependent on them. The literatures after the year 1960 have pointed out the significance of setup time and cost in static and deterministic problems of scheduling [24]–[26] and [27], [28] .

Cheng et al. [29] have made a review on scheduling problems with regard to flow shop, whereas [30] surveyed the same problem with respect to batching. Allahverdi et al. [31] have offered a widespread literature review comprising settings for dynamic and stochastic problem in various shop atmosphere such as, only one machine, many machines operating simultaneously (parallel), flow shops, and job shops. It is evident from the survey of number of publications in the field of "scheduling problems" that scheduling is an integral part of FMS and more innovations are needed for obtaining better FMS. The innovation may be in the form of adding many features that are impacting the solution quality of FMS in the objective function or it may be a novel algorithm combining many iterative/ heuristic/ neighbourhood algorithms to provide a better solution. Economic aspect like setup time/ cost has got a tremendous influence on the profit of the manufacturing firm in terms of saving [32]. The two main components in how an FMS operates are job and tool flows. A variety of topics directly connected to the flow of work pieces have been the focus of research using the parts mobility approach [6]. In addition, the scheduling of machines and vehicles has received a great deal of focus from academics as separate issues. But very few academics have stressed the significance of concurrently scheduling machines and AGVs. This work will focus on such arena in view of improving the performance of FMS with effective scheduling.

1.10 RESEARCH GOALS

The basic purpose of this study is to plan and execute a multi-objective scheduling method in flexible manufacturing systems using meta-heuristic techniques.

The key objectives of this research are**,**

- To frame a multi-objective mathematical model for increasing the performance of an FMS.

- To examine the optimum solution for the sequencing problem related to scheduling or deciding the optimal sequence by checking the entire possibilities.

- Solve multi-objective scheduling of FJSP for numerous benchmark instances with the following objectives for minimizing
  1) The maximum completion time of machines or jobs makespan (MS_{max}).
  2) Maximum machine workload (WL_{max}), i.e., the maximum processing time of a machine.
  3) The total workload of machines, i.e., total processing time over all machines (WL_{total}).
  4) Total idle time, i.e., idle time of machines (T_{idle}).
  5) Total tardiness, i.e., lateness of jobs (T_{late}).
- Minimize the machine idle time and the total penalty cost by a combined objective function (COF) for an FMS setup.
- Obtain an optimized schedule with an objective to minimize makespan, maximum machine workload and total workload for a case study conducted in an oil blending plant in Faridabad.
- To develop three novel hybrid algorithms 1) GAPSOTS (Hybridization of GA, PSO, TS) 2) HAdFA (Hybrid Adaptive Firefly Algorithm- FA with adaptive features hybridized with Simulated Annealing-SA) 3) HFPA (Hybrid Flower Pollination Algorithm which is the hybridization of the "Flower Pollination Algorithm with SA") for solving the aforementioned objectives.

- Study the impact of adaptive parameters in FA and compare it with FPA without adaptive parameters. Very few papers have used adaptive parameters in SI. To this date none of the articles in FA, have incorporated adaptive features with FA for scheduling.

- To compare the performance of the novel techniques developed with other state-of-art techniques such as GA, PSO, simulated annealing (SA) to ensure its superiority.
- To obtain an optimal schedule for concurrent scheduling of AGVs and machines in a FMS setup with the objectives to minimize makespan, mean flow time, mean tardiness.
- The simulation was done on 63 well-known benchmark problems of FJSP, an FMS setup and a real-life case study.

1.11 CONTRIBUTIONS OF THE PRESENT WORK

- This work considers the overall workload of a machine, setup time and the workload of the biggest machine in addition to the makespan (completion time). Thus, the problem is well-defined with multi-objectives.
- A meta-heuristic approach is considered with hybrid algorithm to elude the few drawbacks of individual approach. The hybrid algorithm will include both the Swarm intelligence and population-based algorithms with adaptive parameters. A total of 3 algorithms are developed to enhance the performance of scheduling.
- Also, case studies will be used for analysis. The analysis will be based on the variety of machines and jobs to be performed. All cases will involve more than one machine and more than one job.
- Application of developed meta-heuristics to schedule AGVs and machines combinedly for multi-objectives.

The research framework for the proposed work is given in Figure 1.3

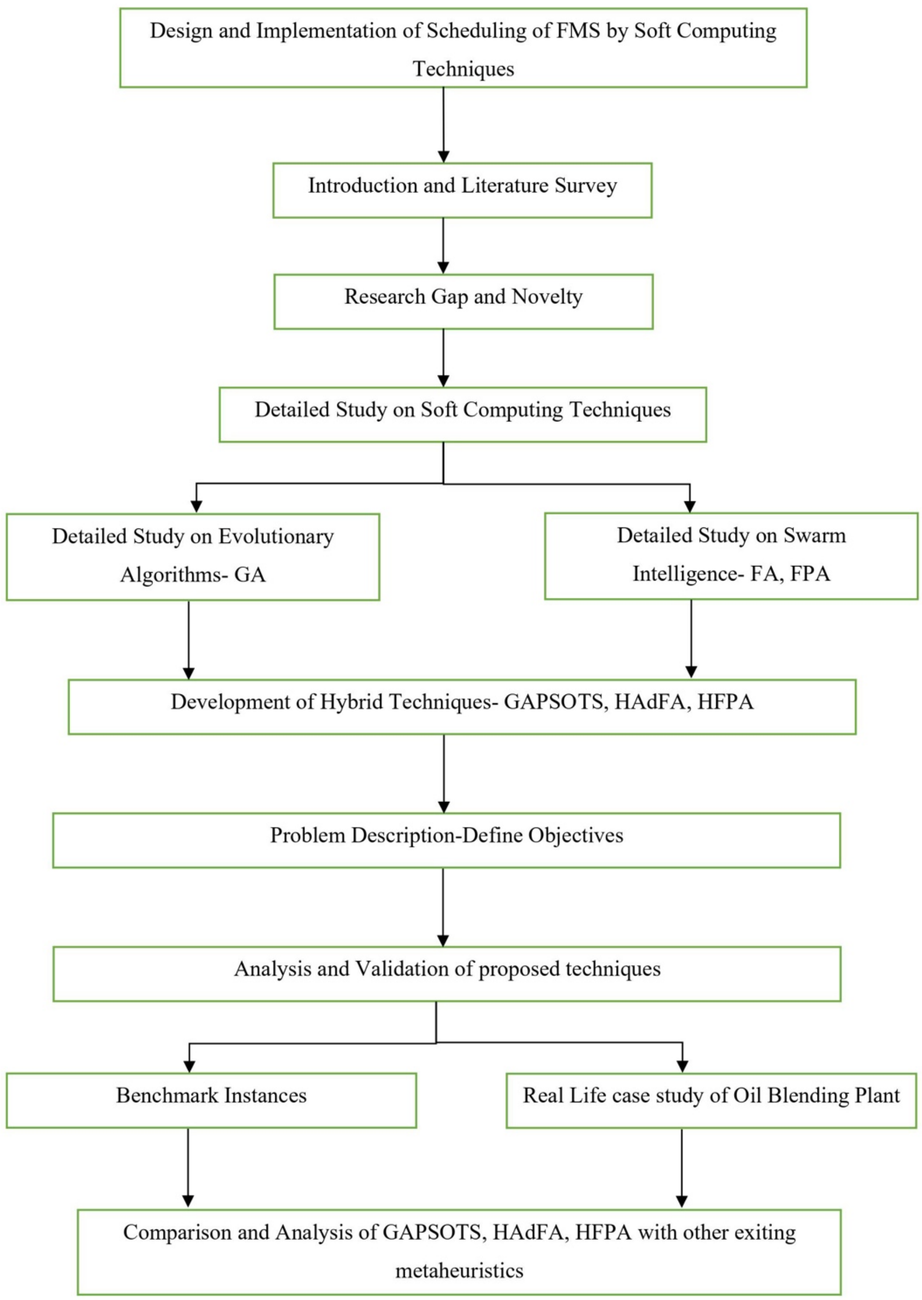


Figure 1.3 Framework of proposed research

1.12 ORGANIZATION OF THE THESIS

The chapter-wise particulars are provided below.

Chapter 1- The first chapter of this research is the Introduction which covers the topic in great detail. In addition, the problem formulation, goals and research objectives and the contributions of the work are spelled out.

Chapter 2- This chapter contains Literature review in which, previous researches relevant to the present study are investigated and deliberated. Furthermore, earlier studies are examined so as to find the gaps existing and incorporate the possible gaps in the present research.

Chapter 3- summarises about Soft Computing Techniques and the proposed three new nature inspired meta-heuristic schemes for various scheduling problems.

Chapter 4- describes about the problem formulation of Flexible Job Shop Scheduling Problem (FJSSP) for multi objectives and the validations of the proposed algorithms are done by comparison of their results with other meta-heuristic schemes and comparison among themselves.

Chapter 5- discusses the combined objective optimization of flexible manufacturing system scheduling for multi objectives and comparison of their results with existing algorithms and among themselves.

Chapter 6- briefs out the problem and analyze the optimal results obtained by scheduling of machines and AGVs combinedly for multi objectives.

Chapter 7- Details the application of proposed algorithms for a real-life case study conducted in an Oil blending plant and analyze the results.

Chapter 8-The conclusion of the work as per the discussions with regard to the aforementioned chapters along with potentials for future work is conveyed in this chapter.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

Flexible manufacturing systems (FMSs) have emerged as an extremely effectual manufacturing scheme during 21st century and helps to yield products with various dimensions and types [33]–[35]. Generally, flexibility is regarded as an interfacing character among a system and its peripherals.

FMS scheduling includes the elements of machine loading, routing of parts, Scheduling the tasks of manufacture, planning and distributing the tools, formation of agenda for utilizing buffer and the plan of AGVs [36]. Many literatures have indicated that the process time of job/ cost and the time for setup/ cost with respact to job are not important. This approximation would make the investigation simple and even echoes on specific applications but it badly influences the quality of the solution with respect to scheduling applications that need the time/ cost of setup concretely.

Inclusion of time/ cost setup in scheduling design has become an integral entity at the middle of 1960s. Mant survey related to this had been performed and their results had been presented for deliberations [28], [29], [31]. The problem of scheduling can be regarded as static or deterministics; researches have been carried out on both models [28], [30].

Cheng et al. [29] have a made a widespread review on the problems of scheduling in flow shop, whereas, Potts and Kovalyov [30] have presented the results of their survey of scheduling relating to batching applications. Allahverdi et al. [31] have offered the outcomes of the review of scheduling as stochastic as well as dynamic problems for various shop applications. The atmosphere for the same has included scheduling with only one machine, many machines grouped in parallel, jop shop, and flow shop, etc. The results from such surveys paved a way for new researches as per the guidelines provided through the results and made them to incorporate the time or cost of setup as a vital factor of scheduling. The tremendous and steady increase of research in this direction is due to the fact that involving the cost of setting up will reflect on the savings and will increase the profit in turn, especially in real-time environments [37].

This literature review chapter aims to provide an overview of the current state of research on FMS, focusing on its design, control, scheduling, optimization, and performance evaluation. The chapter will begin by introducing the concept of FMS and its various components, followed by a review of the literature on the design of FMS.

Next, the chapter will review the literature on FMS scheduling, which involves the determination of the order and timing of tasks to be performed by the various components of the system. This will include a discussion of the different scheduling techniques that have been developed for FMS, such as rule-based and optimization-based scheduling, and the factors that influence the choice of scheduling technique.

The chapter will then review the literature on FMS optimization, which involves the maximization of system performance by optimizing the use of resources such as raw materials, energy, and labor. This will include a discussion of the different optimization techniques that have been developed for FMS and the factors that influence the choice of optimization technique.

2.2 OVERVIEW OF FLEXIBLE MANUFACTURING SYSTEM

The FMS permits to produce a variety of product types for which, the reconfiguration is not required for the entire length of the manufacturing line. In addition, it contains a huge set of numerically controlled machines [38]–[40] with multifunction ability [41], [42], an automatic material handling system [43]–[45] and an online computer network [46]. This network is accomplished of controlling and directing the entire system [47].

The significance of having a rapid turnaround, minimum cost with regard to workers and inventories, and superior quality are the favorable benefits and they make FMS preferable in modern schemes of manufacturing in industrialized segments [48].

The events of manufacturing, tasked related with transportation and storing, etc. are to be correctly scheduled for increasing the efficiency of the FMS in a global manner. The event of scheduling is influenced by various factors such as, automation level, particular FMS characters, the plant at which FMS is situated and its functional rules in addition to the resources of FMS [49]. The growth of scheduling with better quality integrating the entire containing resources of FMS, viz. machines, tools, automated guiding vehicles [50], buffer, is one among the major functional issue to be redressed in such atmosphere [51]. More precisely, effectual scheduling is an indispensable deed that can augment the efficacy and utility of resources [52]. The entering jobs will have recurrent fluctuations in the design of parts, which will impose higher complications to the problem of scheduling. The work concentrates on the schedule of various jobs entering into the system in an effective manner and also aims to maximize the utility of the system with increased system output, especially when the machines are fortified with various tools and periodicals for tools. In such settings, the performances regarding the schedule are described in terms of temporary activity of standard middle term efficacy in quantifying the time of production, delay, etc. All these factors make scheduling a complex one and careful handling is needed through proper architecture for governing the FMS. Normally, the terminology "scheduling" is a vital gismo to engineering and manufacturing and has severe effect on process productivity [53]. The main aim of using scheduling is to diminish the time and cost involved with production and it directs the particular production amenity, the apparatus on which it is to performed and the staff who must work along with that equipment during manufacturing process [54]. The key intention of "production scheduling" is to increase the operational efficiency along with cost reduction [55]. A variety of software products are offered for the models of manufacturing. They take into account the entire process starting from the raw material acceptance, manufacturing to procuring. "Steel-M1" and "Gear-X" are the raw materials procured using which, AX-100,200, BX-100,200 and CX-100,200 are made. The various operations associated in making the above products are cutting, extra processing, assembling, packing and selling [56].

2.3 TYPES OF FMS

The flexible manufacturing system categorized different types depending on the sorts of activity they perform. These are discussed as follows:

2.3.1 Based on Kinds of Operation

FMS can be distinguished dependent upon the types of activities they achieve:

I. **Processing operation:** This task modifies a "work material" beginning with one state to the following state by moving to the final needed part or thing. It upgrades the value by the arrival of the beginning materials or shifting the geometrical properties.

II. **Assembly operation:** It incorporates connecting a minimum of two sections to create another component, which is known as a subassembly or assembly. Unchanging/ unalterable connecting process, which involves brazing, adhesion bonding, extension fits, welding, fusing/ fastening, rivets/ bolts and press fitting.

2.3.2 Based on Number of Machines

With reference to the quantity of machines in the scheme, the standard types of a flexible manufacturing system are discussed as follows:

I. **Single machine cell (SMC).** This comprises of completely atomized machine, which is accomplished unattended activities for a timeframe that is longer than one machine cycle. It is fit for handling, responding to changes in the production schedule, processing different part styles and accommodating freshly entering part while introduced. In such scenario, handling is consecutive, not simultaneous.

II. **Flexible manufacturing cell (FMC).** This encompasses a part handling system and two or three processing workstation. The part taking care of framework is associated with an unload/load station. It is fit for synchronously producing various parts.

III. **A Flexible Manufacturing System (FMS).** This system possesses 4 or more workstations (turning centers or CNC machining centers) for processing and is connected automatically by a distributed computer system and mechanically by a common part handling system. It likewise incorporates non-handling work stations that help generation yet don’t legitimately take an interest in it. For example, coordinate measuring machines, pallet/ part washing stations. These features suggestively distinguish it from FMC.

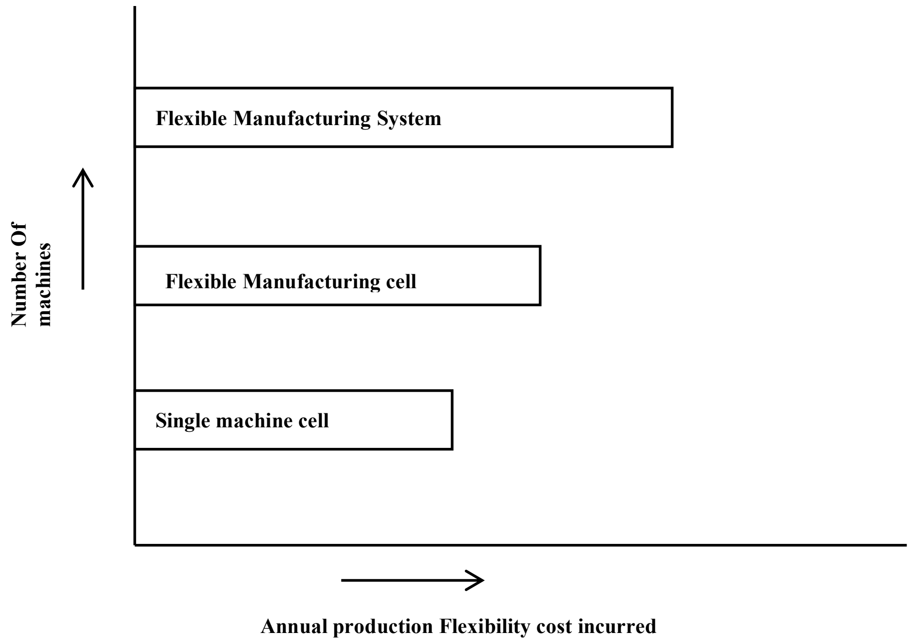


Figure 2.1 Comparison of three categories of FMS

2.3.3 Based on Level of Flexibility

One more taxonomy of FMS is done on the basis of the degree of flexibility connected to the arrangement. Two groups are notable here:

I. **Dedicated FMS:** Several ranges of part styles can be created in this group. The model of the product is assumed stable and hence, the scheme may be modelled with specific number of process speciality to have an effective operation.

II. **Random order FMS:** The extensive fluctuations happening in the part patterns are managed with this group. The degree of flexibility with this group should be more than the previous group in order to lodge the heavy fluctuations occurring. Moreover, this type has potential to process the parts that are highly complex in nature. To accomplish the same, a refined control scheme employing computers is utilized for this category.

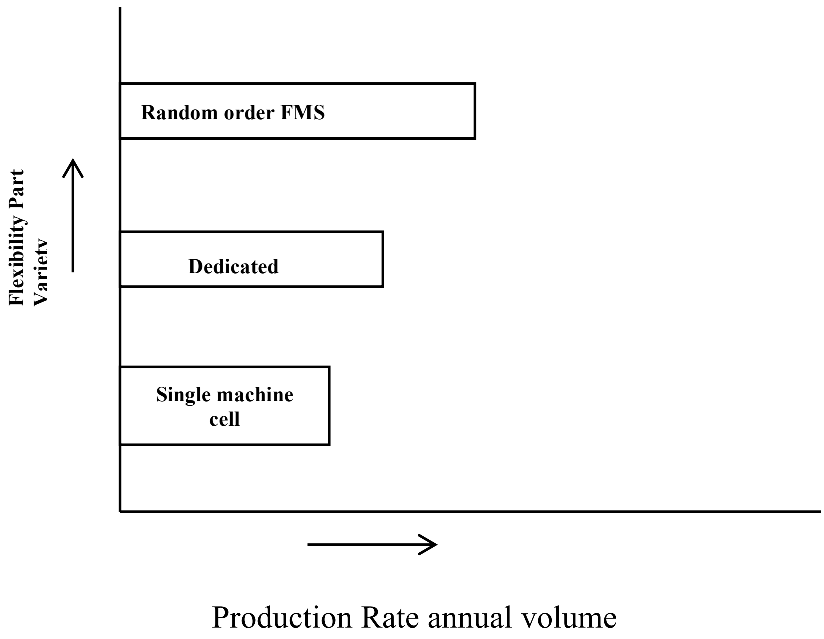


Figure 2.2 The difference between dedicated and random-order FMS types

According to the quoted definition, there exist numerous basic components in FMS. An outline for recognizing the FMS components is provided in the subsequent section. It mainly made up of 2 subsystems namely,

- ✓ Physical subsystem
- ✓ Control subsystem

The former takes account of the elements presented below:

1. Workstations: The components involved here are, Numerical Controlled (NC) machines, machine tools, apparatus for scrutiny/ inspection, loading & unloading action, and area/ space for machining.

2. Storage Retrieval Systems: This function like a buffer for the period of WIP (work-in-processes) and carries devices called carousels, which helps to stock parts for few moments from work stations to operations.

3. Material Handling Systems: This carries powered vehicles/ transportations, conveyors, automated guided vehicles (AGVs), and few more arrangements to transmit parts from one workstation to the other.

Control subsystem has hardware and software elements, which are described as below:

1. Control hardware: It has computers of mini and micro sizes, PLC (controllers working on the programmable logic), networks for communication, devices used for changeover (switching) and other devices connected externally to the system, which may have printers for storing memory for improving the operating capacity of FMS.
2. Control software: This is having a group of files through which, the physically present subsystems are regulated/ controlled. Anyhow, the compatibility among the hardware and software decides the efficacy of FMS.

2.4 PROBLEMS IN FMS

Stecke[57] did a detailed analysis on FMS problems and classified those problems based on “Design, planning, scheduling and control”. Figure 2.3 shows the different types of FMS problems at various stages of installation of FMS.

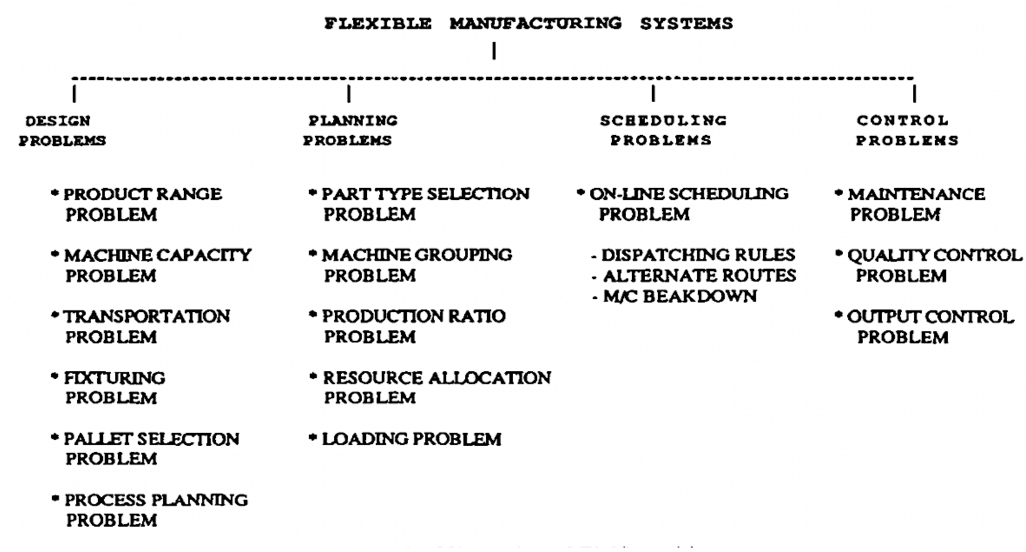


Figure 2.3 Problems of FMS

2.4.1 Importance of Scheduling in FMS

Among various problems encountered in successful implementation of FMS, Scheduling plays a pivotal role. The scheduling is an important “decision -making” process. Effective production scheduling is important for several reasons. Firstly, it ensures that the required products are produced at the right time and in the right quantity, which is critical for meeting customer demands and maintaining customer satisfaction. Secondly, it helps to optimize the use of resources, such as machines, materials, and labor, by ensuring that they are used efficiently and effectively. Thirdly, it helps to reduce lead times and inventory costs by ensuring that production runs smoothly and that there are no delays or bottlenecks in the production process. [58] [59]

Ojstersek et al [60]classified the scheduling problems of FMS into three main categories. Table 2.1 shows the categorization of FMS problems.

Table 2.1 Categorization of FMS problems

Shop Environment	**Job Characteristics**	**Optimality Criterion**
Job Shop Scheduling (JSP)	Pre-Emption Condition	Makespan Related • Makespan • Completion Time • Maximum Machine Utilization
Flexible Job Shop Scheduling Problem (FJSSP)	Precedence Relationships	
Dynamic Job Shop Scheduling Problem (DJSSP)	Processing Time	Due-Date Related • Total Tardiness • Total Earliness • Number Of Late Jobs
Flow Shop Scheduling Problem (FSSP)	Release Dates	
Open Shop Scheduling Problem (OSSP)	Due Dates	

Several models and methodologies have been designed with different algorithms for solving the issue of planning and scheduling of FMS. Over the decades researchers implemented Exact methods, Heuristics and recently Meta-heuristic approaches. The meta heuristic approaches are further classified into Evolutionary based algorithms (EA) and Swarm intelligence-based algorithms (SI). Depending upon the number of objectives to be solved, the FMS scheduling problems can be further divided into single objective and multi objective problems. Following sections briefs about the different techniques implemented for various objectives and FMS problems.

2.5 LITERATURE STUDY OF FMS - PROBLEM TYPE, OPTIMIZATION TECHNIQUES.

2.5.1 Studies related to Design, implementation of FMS

Stecke [57] provided a detailed report on “design, planning, scheduling and control problems” encountered in FMS. This provides a good base paper to understand the basics of Implementation of FMS.

He et al[61] studied the sequencing and scheduling of parts in flexible manufacturing systems in a “mass customization/mass personalization (MC/MP) environment”. They analysed the input scheduling rules for robots and machines.

Negahban and Smith [62] did an exhaustive review for design and operation problems encountered in FMS. They also analysed various simulation packages utilized for FMS by the researchers.

Freitag and Hildebrandt [63] designed a simulation environment for a real case study of semi-conductor manufacturing problem. They used a “multi-obejctive GA” for due date related objectives.

A Study by Krishna et al. [64] had made a scheduling for FMS that involves a problem of optimization of combination type and the complex level is higher. As the management of material is a vital constituent of FMS, the scheduling of treating the material is syndicated to machines’ schedule. The authors have incorporated the impact of buffer in deciding the FMS output by means of simulation along with the regular objective of lessening the makespan. A “Flexsim” simulation package has been applied for validation purpose, which is a discrete event simulator. The scheduling mode for the two objectives is made parallel instead of sequential manner. For analysis, a typical problem comprising various layouts and dissimilar job groups are taken.

A study by Gang and Quan [65] has described a multifaceted arrangement for FMS and has highlighted the trouble in planning for production. An all-purpose FMS scheduling model is constructed and a list algorithm has been suggested depending on multiple levels/ degrees of flexibility and this is the central part of the suggested algorithm. Then a common scheme for planning is evolved. The proposed model has been investigated using the suggested algorithm and validated by means of planning layout. The precision and compliance are then verified with reference to the model as well as the algorithm.

A work by Mallikarjuna et al. [66] has considered a multi-objective optimization approach toward scheduling FMS that has to be optimum manner and considered to be one of the constraints in constituting the ladder type layout through different metaheuristic approach like SA, Particle swarm method (PSM) and so on. The outcome of objective function with regard to the total of iterations performed and computational time. The SA and SM were tested by considering the different Ladder layout issues. The simulated result has been compared with different optimization technique like SA and PSM. Finally, they concluded that the obtained solution using SA is better than the PSM but the computational time of both approaches is almost same.

Erdin and Atmaca, [67] have modelled and executed the FMS for computing the desirable quantity, usage, and the order of sequence of various workstations and the layout for the specified number of several production parts, timing and sequence to process. The investigation and computations are performed by means of analytical model employing bottleneck principles and clustering methods that employs the ordering through ranking. The parts those have resemblances of each other are made as manufacturing units for making the analysis easy and also to utilize the workstations effectively. The derived results of the proposed model have been compared with traditional models of manufacturing for evaluating the efficiency.

Liu, et.al. [68] have adopted support vector machine technique for the scheduling of FMS intended to allocate the rules of dispatching and ultimately realizing better performance. The impulsive and erratic conditions of dynamic nature like various types of parts, their mix-ratio, and influx of jobs. The SVM is regarded as a superior learning model than others because of its generalized performance with multi-kernel environments. To substantiate the results through simulation, the classical model of FMS with physical layout is considered. The SVM is operated with radial basis function kernel and it has been identified from the results that the SVM outperforms the traditional models. Also, the setting-up process with SVM has taken very less time.

Lee and Ha [69] have presented a genetic approach to identify the pseudo optimal value for the process of integrated plan and scheduling issues. They have considered the integrated problem as a combinational category of optimization with NP-compete type. Thus the study targets to get a solution for process plan and its schedule at a time. The intricacy associated with such issues is more as it involves or necessitates a number of flexibilities and limitations for FM environments. In order to tackle them, the prevailing studies have left out particular flexibilities and restraints or else those have constructed an algorithm with composite structure. More precisely, the genetic approaches have been enforced to build several chromosomes to include several flexibilities. This definitely would augment the complexity of the algorithm, which in turn, deteriorates the performance. The suggested novel algorithm has an integrated chromosome description and unites several flexibilities into just one string. So, it is possible to adapt a simple and regular procedure of genetic approach and the formerly made genetic operators. Experimentations conducted on several standard problems have proved that the suggested model has upgraded the makespan by around 17% with minimum time of computing.

Malik and Pena [70] have established the application of model checking for the task scheduling with optimum values in FMS. The scheme has been though of a discrete event system and the minimum restricted safety behaviour is amalgamated as per the supervisory model of control theory. The time restraints are included to the model as finite state machineries in the extended form. The Supremica, which is a tool of model checking and the discrete event systems are applied to compute the optimum time for scheduling.

Research by Baruwa and Piera [71] developed a coloured Petri net (CPN) for identification of optimal solution for problems associated with scheduling. This research uses the CPN approach incorporates reachability graphs for finding a solution. Further, this research focused on reducing memory requirement it exploits structural equivalence graphs with desire reachability in flexible manufacturing systems' (FMS). Results illustrated that developed approach performs effectively rather than existing search methods adopted for scheduling for large size FMS.

A study by Lee and Lee [72] described efficient scheduling for the schemes of manufacturing with flexibility, using a new heuristic function depending on T-timed Petri net. The suggested functions help to diminish the make span effectively as compared to the available functions with regard to the desirable amount of states and time utilized for computing. These functions of the proposed scheme have ensured that they are more permissible and knowledgeable than the conventional “resource-cost reachability matrix”. When the heuristic functions are enhanced, it is even possible to obtain the initial adjacent optimum solution quickly. Moreover, a heuristic function of adaptable version for entire states has been recommended [73]. The experimentation has been carried out through an arbitrary problem generator and the outcomes infer that the recommended scheme has performed well as estimated. Research by Zambrano Rey et al.[74] proposed a semi-hierarchical architecture for optimization of flexible manufacturing system (FMS) present scenario. This research considers various myopic decisions through optimization technique. With optimization mechanism in simulation local and global calculation is evolved for FMS. Simulation analysis of the proposed approach performs controlling of assembly cell by means of higher hierarchical approach. In the myopic reduction of behaviour variance of completion time is measured as performance measures. Through simulation analysis, it is concluded that semi-hierarchical architecture reduces behaviour of myopic in FMS with the increased ability to offer a balance between disturbance ability and low complexity maintenance; hence it will be appropriate for production control. But this research fails to provide an appropriate mathematical calculation for proposed approach.

2.5.2 Studies related to Flexible job shop scheduling problem (FJSSP)

Mastrolilli and GAmbardella [75] proposed that the neighbourhood functions are effective in improving the solution quality for the flexible job shop problem, and that the selection of a suitable neighbourhood function is an important factor in achieving good results with metaheuristic algorithms. They developed metaheuristic algorithms such as simulated annealing and tabu search. Neighbourhood functions define a set of possible solutions that are "close" to the current solution, and the metaheuristic algorithm explores these solutions in order to find an improved solution.

Mati et al [76] proposed a heuristic approach of integrated greedy algorithm that combines a priority rule for job sequencing and a dispatching rule for machine assignment. The priority rule determines the order in which the jobs are processed, while the dispatching rule assigns each job to a machine based on its availability and the processing time of the job.

Xia and Wu [77] suggested a method with hybrid optimization strategy that combines a local search algorithm and a “multi-objective genetic algorithm (MOGA)”. The local search algorithm is used to enhance the quality of the solutions produced by the MOGA. The MOGA is used to develop a diverse set of candidate solutions that represent various trade-offs between the competing objectives. Additionally, the research suggests a novel neighborhood structure for the local search algorithm that is adapted to the MOFJSP's properties.

Motaghedi-larijani et al [78] suggested a method employing a "multi-objective genetic algorithm (MOGA)" to develop solutions that simultaneously optimise many objectives, like lowering makespan (total completion time) and the quantity of late jobs. In addition, the research suggests a heuristic initialization technique that yields high-quality initial solutions for the MOGA as well as a novel fitness function for the MOGA that integrates both the makespan and tardiness objectives.

Xing et al [79] developed an efficient integration between the knowledge model and the "Ant Colony Optimization (ACO)" model is provided by the "KBACO" algorithm. The "KBACO" algorithm's knowledge model extracts some previous information from ACO's optimization and then utilises the information to direct the present heuristic searching**.**

Khadwilard et al [80] describes the implementation of the proposed Firefly Algorithm approach and investigates the impact of different parameter settings, including the number of fireflies, the light absorption coefficient, and the attraction coefficient, on the performance of the algorithm. The paper also proposes a novel initialization method that generates high-quality initial solutions for the FA.

Research by Bhushan and Kumar [81] focused on improving job scheduling in FMS for dynamic variation in incoming jobs and efficiency. For optimization of jobs in FMS, this research uses Taguchi philosophy and genetic algorithm. The designed optimization approach schedules various incoming jobs within the system effectively with an increase in system throughput and utilization of machine which is incorporated with a distinct number of tools and magazines. This research concluded that experimental analysis of simulation for real-world system is certain drawbacks such as higher cost, dangerous and time-consuming this are all overcome in this proposed approach. However, this research does not offer information about tools considered for this research.

Karthikeyan et al [82] developed a novel strategy approach based on the combination of the data mining (DM) technique with the "particle swarm optimization (PSO)" algorithm. PSO is employed to assign tasks and choose the sequence in which jobs are processed by machines. By using data mining to extract knowledge from solution sets and identify close to optimal solutions to combinatorial optimization issues.

Yanibelli and Amandi [83] takes into account two competing, top-priority optimization goals for project managers. One of these goals is to shorten the project's timeline. The best group of human resources should be tasked with carrying out each project activity, according to the other goal. A multi-objective hybrid search and optimization technique is suggested as a solution to the issue. This approach combines a multi-objective evolutionary algorithm with a multi-objective simulated annealing algorithm. To enhance the efficiency of the evolutionary-based search, the multi-objective simulated annealing algorithm is incorporated into the multi-objective evolutionary algorithm.

Roshanaei et al [84] suggested to use a meta-heuristic that combines the "Artificial Immune and Simulated Annealing (AISA) algorithms". The optimization gap is assessed by contrasting the unsatisfactory AISA solutions with their MILP exact optimal counterparts found for small- to medium-sized F-JSSP benchmarks. Mould and die shop case study was used to study the efficiency of proposed algorithms.

Yuan et al [85] developed "hybrid harmony search (HHS)" to solve FJSSP. The continuous harmony vector is first converted using developing approaches into a kind of discrete two-vector code for the FJSP in order to make the harmony search (HS) algorithm adaptable to the FJSP. Second, by effectively decoding the converted two-vector code, the harmony vector is mapped into a workable active schedule, which may significantly condense the search space. Thirdly, to ensure that the initial harmony memory (HM) occurs with a specific level of quality and diversity, a successful initialization approach combining heuristic and random procedures is presented. The algorithm was tested on benchmark problems.

Buddala and Mahapatra [86] developed "Teaching–learning-based optimization (TLBO)" for minimizing makespan by an integrated approach. They developed a new local search method influenced from GA and tested the algorithm on various benchmark problems.

Li et al [87] developed an adaptive evolutionary algorithm where they used a new encoding for population initialization. They made changes in other genetic operators too. Their focus was on parameter adaptive settings and the testing results proved that adaptation of parameters increases the efficiency of FJSSP.

Ning et al [88] developed "quantum bacterial foraging optimization" for objective of low carbon emissions. They analyzed the results with ANOVA and demonstrate the efficiency of algorithm.

Jiang et al [89] implemented a "discrete animal migration optimization (DAMO) to solve the dual-resource constrained energy-saving flexible job shop scheduling problem (DRCESFJSP)" to minimize the amount of energy used overall in the workshop.

"A dynamic self-learning artificial bee colony (DSLABC) optimization technique" is developed by Long et al [90] to resolve the DFJSSP, which is the issue of adding new jobs into flexible job-shops. Initially, the "self-learning artificial bee colony (SLABC) algorithm" is created by combining the "Q-learning algorithm" and the conventional "artificial bee colony (ABC)" method. The update dimension of each iteration of the ABC algorithm can be dynamically modified using the learning features of the Q-learning algorithm, which increases the convergence accuracy of the ABC. Second, the precise dynamic scheduling technique is chosen, and the DSLABC is suggested. The activities that have not yet begun processing will be postponed together with the newly inserted job.

2.5.3 Studies related to Scheduling of AGVs and machines in FMS

Pandey and Singh did a review of "automated guided vehicle design and control". The report presents a technique to integrate several lines of AGV research and makes recommendations for future work on the most important associated issues, such as vehicle scheduling. In various job shop setups, scheduling issues involving vehicle dispatching, guide-path design, and routing are resolved.

Chawla et al [91] developed "Modified Memetic Particle Swarm Optimization Algorithm (MMPSO)", which combines "Particle Swarm Optimization (PSO) and Memetic Algorithm (MA)", is used to schedule multi-load AGVs with the least amount of travel and waiting time possible in the FMS. The proposed "MMPSO algorithm" demonstrates balanced exploration and exploitation for the global search method of the traditional Particle Swarm Optimization (PSO) algorithm and the local search method of the Memetic Algorithm (MA), which further yields efficient and effective initial feasible solutions for the scheduling problem for multiple load AGVs.

Reddy et al [92] addressed the simultaneous scheduling problem of AGVs with machines and tools using a flower pollination algorithm (FPA) based on the pollination process of flowers and a nonlinear mixed integer programming (MIP) formulation to describe the joint scheduling of machines, AGVs, and tools.

Zheng et al [93]minimized the makespan while scheduling machines and AGVs simultaneously. They proposed "Tabu search" for this study. The suggested approach incorporates the production of two neighbour solutions, which are then alternatively and repeatedly employed to improve solutions, as well as a novel two-dimensional solution representation. Also, for the large-size problems, a better lower limit computation method is introduced.

Prasad and Rao [94] proposed a "black widow optimization algorithm" to solve machines and tool scheduling side by side using many machines in a flexible manufacturing system, the best possible sequences are generated to reduce production time. The objective was to minimize makespan.

Research by Sreenivas et al. [95] performed parallel scheduling using a simulated annealing algorithm for addressing parallel scheduling of AGV's and machines. Based on simulated annealing trajectory-based method is developed. A focused approach aimed to identifies minimum makespan, mean makespan and tardiness of the system. Simulated annealing is tested for over 20 standard problems collected from analysis of literature. The developed annealing is considered the adoption of two-stage calculation performance. The limitation observed in this research is mathematical formulation or derivation is minimal for tardiness, job completion time and mean value.

Nageswara rao et al [96] research aims at the utility of concurrently scheduling two similar automated guided vehicles (AGVs) and machines in a flexible manufacturing system (FMS). AGV performance optimization is essential for enhancing FMS efficiency. For the purpose of providing the best Sequence with relative makespan value and AGVs schedule for ten work sets and four lay outs, a hybrid meta heuristic method is designed and created with Java code.

A work by Singh and Jayant [97] presented a problem in appropriate selection of FMS in specifically for a manufacturing organization. In the process of FMS decision-making process, three criteria are considered such as ELECTRE III, VIKOR and improved

PROMETHEE for decision-making system. Based on the numerical expression is presented and discussed three multiple criteria decision-making process of FMS. Ranking of three selected variables are comparatively examined with VIKOR, improved PROMETHEE and ELECTRE III provide the good alternative set value of {7, 4}. Alternative ranking values are higher for 7 and 4 with the alternative of 7 which is ideal according to the RSC, IQ, FSU and closet criteria according to ideal criteria IMR, CMC and closer based on the best alternative of 7. This research concluded that matching method would be applied for the appropriate class of problem required. Validation procedure is adopted, and feasible application is explored. To resolve problem researcher are struggling to select guide for efficient method by means of the theoretical and practical formulation. But this research fails to provide appropriate steps for the entire research process and techniques are not developed.

Research by Saren et al. [98] focused on effective performance characteristics of decision-making strategies involved in FMS by means of hierarchical structure. This research discussed the tools, parts and machine for the transition. This research considers the FMC model formulated on laboratory operation of Faculty of Managerial and Technological Engineering, University of Oradea. Analysis is performed with consideration of arrival rate of part and decision processing of parts through transition of each sub-model. Analysis of hierarchical model is performed through CPN tools software for FMS modelling and formulation. Results illustrated that implementation of transition and place is based on arrival time and processing time of parts through hierarchical model. This research stated that various parts are processed through different sub-models which increase overall performance of FMS. However this research does not provides a clear conclusion for this research and mathematical derivation for the proposed model also.

A study by Gothwal and Raj [99] developed a framework for managing task proactively for FMS. To retain best solution this research integrates simple additive weighting and weighted product scheme along with the process of analytical hierarchy. Weights are determined and measured with AHP using SAW and WPM approaches with management of performance measure prioritizing with improved decision-making performance. For conversion of qualitative to quantitative measures fuzzy logic is adopted in this research.

A study by Chawla et al. [100] investigated dynamic job selection for simulation by means of rules involved in dispatching and scheduling is performed with consideration of multi-load scenario of automated guided vehicles (AGVs) FMS of different size vehicles.

AGVs incorporates multi-load scenario consideration of rules for dispatching machine initiated nearest vehicle (NV) and materials for pick and drop in FMS. Results revealed that rules for dispatching involve similarity of outperforms of jobs and rules involved in dispatching. This research concluded that FMS throughput depends on AGV speed and fleet. But this research does not consider any mathematical formulation for analysis in FMS.

A study by Erol et al. [101] have proposed a method to machines and AGVs inside the scheme of manufacturing by means of multiple agents dependent scheduling. The presented method operates through a practical atmosphere and creates viable schedules with the help of bidding or negotiating mechanisms amongst the agents. Furthermore, they suggested investigating and additional refining the multi-agent systems (MAS) methodology for resolving scheduling in a dynamic ambience [102] and controlling the issues/ complications in manufacturing.

Burnwal and Deb [103] have implemented an iterative model using cuckoo searching (CS) algorithm for optimizing the process of schedule in FMS. It is accomplished in two folds: by reducing the cost of punishment (fine) because of the postponement in manufacturing and by increasing the utilized time of machine. It is intended to illustrate the use of CS related structure to identify the optimum jobs. The Levy fight operator has been used for this purpose by making minute changes in the suggested algorithm as it is much suited for applications of unconnected nature.

A Study by Huang et al. [104] has proposed efficient scheduling for FMS and has estimated an upgraded searching tactic and its usage with FMS scheduling in the P-timed Petri net context. It is possible to apply the acceptable and non-acceptable exploratory/ heuristic functions at a time using recommended algorithm by implementing the Petri net. It is also established that the resultant heuristic function of combinational type is permissible and highly conversant than the other elements. The investigational results for a sample FMS and numerous groups of arbitrarily created issues demonstrate that the presented searching technique does well and has matched the beliefs.

Research by Başak and Albayrak [105] developed a Petri net (PN) model for a FMS. An analysis is evaluated with consideration of PN design and implementation mechanism in FMS. Modelling of FMS tools adopts PNs with principal modelling with the implementation of object-oriented Petri nets (OOPNs) method for design plus implementation of effective controlling for production. Developed PN utilizes timed marked graph (TMG) which is generally a subclass of PNs. The conclusion of the research stated that real-time scheduling and modelling effectively controls the FAMS (flexible automotive manufacturing system) located in Valeo Turkey. Since this research focused on performance evaluation of Turkey alone, it will not be significant for other countries.

Research by Chawla et al. [106] investigated the rules associated with the dispatching of dynamic job selection approach for multi-load automated guided vehicles (AGVs) in FMS have different sizing. Sizing related to AGVs are vehicles initiated through the nearest vehicle (NV) involves certain framed rules for dispatching and material drop for flexible manufacturing system (FMS). For consideration of two size in facility of FMS is evaluated with job selection for five different jobs with desire rules in dispatching. Rules related to dispatching in FMS are evaluated using simulation process based on the consideration of selection criterion for jobs similarity for the destination. Through analysis, it is concluded that FMS throughput increases or decreases with consideration of AGV speed and fleet size. This research does not offer any mathematical explanation or simulation measures for FMS scheduling.

A study by Kumar et al. [107] has examined the problem in scheduling associated with FMS by the application of meta-heuristics approach improvement in planning for production for FMS scheduling. This research considers flexible manufacturing system 6 machine producing system with consideration of 3 distinct parts with 3 machines at different setup considers each setup and consideration of 3 alternative routes. Scheduling for optimization involves Bacterial Foraging optimization algorithm (BFOA, GA and Differential Evolution (DE) for an optimal scenario in to consideration. Through analysis, this research concluded that proper decision in industry performs excellent setup and scheduling by the meta-heuristics approach and modelled using Pro model software for different runs. This research does not provide application variation of FMS sector.

A study by Mousavi et al. [108] developed a mathematical model by integration of genetic approach, swarm based approach (PSO), and hybrid GA-PSO for optimal job scheduling with AGVs for reducing makespan and quantity of AGVs for consideration of charging the battery. Numerical analysis of results demonstrated that makespan is decreased by application of three algorithms through scheduling before and after optimization approach for AGVs. Results illustrated that GA-PSO provides optimum results and performance output for the other two algorithms. AGVs mean value for operation improves efficiency in terms of 69.4%, 74%, and 79.8% for a multi-objective function with all three algorithms mentioned in the proposed model. Validation and evaluation of the model is evaluated through Flexsim software for simulation performance measures. This research fails to provide an implementation of proposed algorithm in the scheduling of AGVs.

Research by Mishra et al. [109] evaluated the FMS manufacturing system for the production of multiple products which requires reconfiguration in production line. This research utilizes Particle Swarm Optimization (PSO) with multi-objective scheduling process for a manufacturing process for tasks such as transport and storage requires appropriate scheduling mechanism. To resolve scheduling problem related to FMS PSO provides an optimal solution for obtaining a solution by means of scheduling by genetic algorithm (GA). But this research fails to provide details about consideration of job for scheduling and processing involved in FMS.

Research by Kamatchi and Saravanan [110] suggested a modified discrete firefly algorithm for flexible open shop scheduling issues are investigated along with five objective functions toward met real-world production situation. Further, they have applied ordinary firefly algorithm, which dealt with an isolated form of the non-stop function like distance, movement and attractiveness for finding correct location of Firefly. The demonstrated outcomes confirm that the suggested approach provides improved results in comparison to the other algorithms specified in research papers. Upcoming scholars may concentrate on multi-objectives with few more objectives that are not included in the present study or the fuzzy version of that.

MATLAB possess strong power of computation and helps to solve problems that are complex in nature. The “parallel computing” tool box helps to perform the computations quickly and effectively when more machines are operated in parallel in FMS. As the modern scheduling has many objectives and some of the objectives are contradictory in nature, such kind of issues should be solved with a powerful tool. Hence, MATLAB can be used for scheduling problems in flexible manufacturing schemes with multiple objectives. Many literatures have reported the application of MATLAB in FMS especially for scheduling [111]–[116].

Research by Rashmi and Bansal [117] developed an ant colony optimization technique for multi-objective task scheduling for Auto Guided Vehicle (AGV) in FMS. ACO adopts effective scheduling path for AGV task scheduling by balancing the objective function with minimized task time of AGVs in FMS. Optimum scheduling is performed through a combination of objective function. Analysis demonstrated that ACO technology is a promising technique with effective dynamic performance capability. Minimal change in objective function improves task scheduling performance which resulted in AGV utilization and overall system efficiency.

Research by Mehrabian et al. [118] has presented a mathematical programming model based on a two-objective function which integrates AVGs routing in the flexible manufacturing system and scheduling of flower shop. Uncertainty is always of higher demand for real-life problems like processing time and due dates. In order to resolve realistic problem, several parameters are considered in this research for the developed mathematical model. In the formulation of a fuzzy based model for programming effective technique is identified through literature. A mathematical model is evaluated using f Non-dominated Sorting Genetic Algorithm-II (NSGAII) and particle swarm optimization with multiple objectives for evaluation of accuracy value and efficiency of the system assessment. But this research does not provide practical application of the proposed model in FMS scheduling for AVGs.

Udhayakumar and Kumanan [119] have aimed to make active scheduling and ordering sequence of job in an optimal way along with a proficient tool that decreases the makespan of FMS. A novel ant colony optimization (ACO) algorithm has been utilized for meeting the objective. The said algorithm tries to get adjacent solutions with optimum values by means of Giffler extension and Thompson algorithm for arriving at a schedule, which is active as well as optimal. The results of the designated algorithm has been compared with the algorithms suggested in earlier researches and the investigation proves that the suggested method helps in achieving improved solution with minimum span of time.

Mathew and Saravanan [120] have presented a genetic algorithm for integrating flexibility in the modern manufacturing schemes. They have accounted various objectives such as, reducing the machines' idling time, decreasing the total cost expended towards penalty, which is imposed for not satisfying the timelines very frequently. Software is established for obtaining the order of sequence of the operations involved in FMS with optimal values. The authors have tested their proposal with 16 tools integrated in a total of 16 numeric control machines governed through computers to process 43 different products. The problems are even experimented through the meta-heuristic approaches pointed out in the research papers. The results of them are matched to the suggested method. The GA approach of considered problem with designated objectives has yielded a globally optimal scheduling once 1700 generations have been completed.

Kumar et al [121] have validated the usefulness of genetic and differential evolution schemes for schedule of FM system. They have used sixteen machineries along with 43 jobs of flexible manufacturing to test the proposed method and the results attained have been used to compare with traditional rules of schedule. The coding for designated approach has been carried out in MATLAB environment of version 7.1.

Nidhiry and Saravanan [122] have utilized tools of CNC machines of 32 numbers in the schedule of FMS that deals with 40 various products. They have decided to deal with 2 different objectives that are opposing in nature. They are minimalizing the idling time of machines and diminishing the overall cost incurred by the way of penalty. They have devised an optimization technique using NSGA-II software, which is capable of handling several objectives at a time. The optimal sequencing of products has been carried out by .net programming setup. The comparisons have been carried out by matching the recommended method with other famous algorithms available in the research papers. The finest solution has been attained after 3000 generations.

Marichelvam et al. [123] have offered a hybrid monkey searching algorithm for resolving the schedule problem associated to flow shop jobs. This problem is a combinatorial problem and NP-hard category due to its complexity. The considered objectives are the reduction of make-span to greater extent and reduction of overall flow time. Here, 2 various sub populations are created with 2 goals and dissimilar rules of dispatch have been utilized to enhance the solution excellence. The performance has been compared with different popular algorithms that are proven to be effective. The results of experimentation and comparison infer that the monkey searching has produced a loftier output to meet out the said objectives.

2.5.4 Studies on Firefly algorithm and Flower Pollination algorithm on other domains

Fister et al [124] provided a comprehensive review on firefly algorithms with its variants. This provides as a good base pare to understand and study about the different adaptations that have been made to FA. This study strongly recommends that FA is good for solving any engineering problems due its simplicity.

"A probabilistic Energy Management system to optimize the operation of the Micro-Grid (MG) based on an efficient Point Estimate Method (PEM)" by an "Adaptive Modified Firefly Algorithm (AMFA)" to achieve an optimal operational planning with regard to cost minimization is developed by Mohammadi et al [125]

Álvarez-Gil et al [126] solved a make to order production unit for minimizing makespan total workload; and maximum workload by a Discrete FA. Customers can choose from a few qualifying product attributes in the proposed framework, which are viewed as the several manufacturing procedures that make up each job.

"An adaptive hybrid evolutionary firefly algorithm (AHEFA) for shape and size optimization of truss structures under multiple frequency constraints" was studied and analysed by Lieu et al [127]. "Differential evolution (DE) algorithm" was hybridized with the firefly algorithm (FA).

Gandomi et al [128] developed FA by introducing chaos into the algorithm. This lead to the effective global search and the algorithm was tested on several benchmark instances.

Yu et al [129] [130], [131]developed various "self- adaptive FA" in order to avoid the FA falling into local optima due to its fixed parameters. Here the authors have suggested to make the parameter change dynamically at each iteration.

Diab and El-Sharkawy [132]conducted a review on recently developed Flower pollination algorithm (FPA). They analysed the applications of FPA, its variants and also added the future scope of the FPA.

Alyasseri et al [133] did a detailed study on various modifications done to FPA till date. And they found that the modifications help the FPA to achieve better results and simplicity of coding is desirable. To deal with the complicated nature of optimization problems, modification, hybridization, and parameter-tuning of FPA are used.

Cui and He [134] introduced "Cat Fish Mechanism" in FPA to weed out the worst solutions obtained during the iterations. Because of this modification the FPA was able to maintain superior diversity in population thereby leading to achieve best optimal results.

Qu et al [135] developed "hormone modulation mechanism (HMM-FPA)" for a "no-wait flow shop scheduling problem (NWFSP)" with an objective to maximize consummate time. The authors introduced a self-adaptive neighbour search for local search. Benchmark instances were used to test the algorithms efficiency and it proved to be better when adaptive parameters are used.

2.6 INFERENCE FROM LITERATURE STUDY

The following section briefly explains the voids in research, commonality and scope of expansion of various research problems in FMS scheduling categorizing with respect to meta heuristics, Objectives, FMS problems.

2.6.1 Meta-heuristics techniques

Solving the problems by exact method needs a rigorous derivations and calculations involved is more and most importantly the computational time is high especially when the size of the problem enlarges and carries more instances. It is even complicated when NP-hard problems are dealt. In such situations, the time of computing exponentially increases with the problem size. Meta-heuristics methods are very good alternative to such problem as the solution provided is very closer approximation to real solutions and the computational time is drastically reduced. When the scheduling problems are concerned with regard to added flexibility in modern manufacturing, many objectives are to be met with minimum time frame. The problem is composite in nature and includes large instances of problem. Using meta-heuristic will offer a compromise between the quality of solution and the time of computing. This can be fit to the wants of real-world problems for optimizing and this method is not demanding in view of problem formation. That is, there is no rule for framing the objective function and constraints based on the decision-making variables. Hence, this method is well adapted to FMS scheduling issues and offers very good solution in least possible time. Also, a hybrid heuristic approach may provide finest solution than single heuristic in the case of multi-objective problems.

Among meta heuristics, Evolutionary Algorithms (EA) and Swarm Intelligence (SI) based algorithms are dominating the optimization research area. SI is gaining popularity due to its simplicity and very few parameters which are easy to control. Computational complexity is less for SI when compared to EA. Yet, not all SI techniques are effective in scheduling of a FMS. Thus, the need arises to formulate an effective SI technique to tackle the scheduling problem. This can be achieved either by parameter tuning or hybridization of one or more techniques.

2.6.2 Multi-objective scheduling technique

The flexible manufacturing focuses on the customer needs and accordingly has to produce products. To sustain in the market and cope up with the competition prevailing in the market, many constraints are there and depending on these restraints, a clear objective has to be made. In general, many literatures have projected the problem with only one objective, which is the lessening of *make-span* to the minimum possible extent.

Even though satisfying this objective is paramount in invariably all the manufacturing industries/ units, this objective alone will not meet the requirements of the customer. Having introduced automation almost in all firms at least partially, the objective converts into multi-varied and involves many parallel machines. A fine is being levied if the particular job is not over as per the stipulated time and in some cases early delivery poses a problem. Thus, the tardiness and workload of machines, and the number of machines used are becoming significant factors in scheduling problem. Thus, making a distinct objective by combining all the needs is mandatory. Obviously, the objective function turns out be a multiple objective function. Even two objectives are referred as multi-objective problem as per the literature. Yet, for rigorous analysis more than two objectives are needed. Hence, there is a need for pertinent design with a generally modifiable structure suiting to multiple production objectives.

2.6.3 FMS Scheduling problems

A hierarchical approach to solving FJSP subproblems is frequently used. Whereas few studies have been conducted for an integrated approach in which both machine assignment and machine routing are performed concurrently. The integrated technique explores two fields and takes into account both subproblems simultaneously. The variety in the search space tends to rise as a result of this pattern of searching. Therefore, it is crucial to devise a strategy for potentially reducing the problem's complexity by conducting a thorough search in the search space. The scheduling of machines and AGVs has to be done together in order to decrease the production cost and makespan. But very few studies have dealt the problem of simultaneous scheduling. Therefore, there is a significant scope to attempt “simultaneous scheduling of machines and AGVs”

Hence, it is intended to address all the drawbacks deliberated so far and design a powerful hybrid algorithm for scheduling problem in FMS.

2.7 RESEARCH GAPS

The analysis of existing literature related to flexible manufacturing system (FMS) certain research objective is observed for consideration of different scenario which is presented as follows: In order to use an efficient approach of meta-heuristic for the problem of concern in this work, the various gaps or intricacies associated with the existing heuristic methodologies in scheduling of FMS have been found and listed.

1. Several meta-heuristics techniques are evolved for solving scheduling problem in FMS, but effective technique is not presented.
2. The performance is conducted based on formulated research objective domain, which may vary for another process.
3. Minimal change in objective function leads to change in entire processing associated with manufacturing.
4. In terms of Swarm intelligence (SI) approaches, FA and FPA are intriguing methods with a broad range of potential applications for additional analysis and testing, including scheduling. In order to solve the FJSSP problem, FPA is a relatively new technique, and academics have only so far suggested using it to solve the single objective problem. In order to address the multi-objective scheduling problem with hybridization, there is room to broaden the FPA.
5. Recent years have seen the introduction of adaptive elements into an optimization technique's control parameters. This allows the researcher to have control over how a strategy investigates and exploits the solutions, and it also allows for the resolution of issues like local optima trapping. Also, is noted that most FMS scheduling experiments are resolved by combining one or more metaheuristics.
6. Integrated approach to solve the sub problems of machine assignment and sequencing in FJSSP scheduling can be attempted.
7. Parameter tuning of FA can be done for FMS scheduling as the literature has shown that parameter tuning plays an important role to increase the fficiency of the algorithm.

8. Multi-objectives (due date related and time related objectives) to schedule FMS can be attempted since most of the studies focused on single objective of minimization of makespan only.
9. Also, most of the literatures have applied heuristic approach to boost the results and it is seen through the investigation that hybrid method combining more than one algorithm is efficient. This study also aims to develop a meta-heuristic approach hybridizing many relevant and feasible algorithms to handle the scheduling problem in FMS.
10. Many of the studies have used benchmark instances to test the efficiency of the proposed algorithm but very few studies attempted to solve a real life case study for multi-objective scheduling optimization.

2.8 SUMMARY

In this chapter, it has been examined about FMS in consideration of several scenarios in to consideration. For analysis of FMS, performance evaluation is considered based on meta-heuristics approach, decision making with consideration of multi-objective terms, multi-objective scheduling and artificial intelligence technique in the processing of FMS. Analysis of literature presented that scheduling in FMS is performed via consideration of incoming jobs count, operation, task assignment and scheduling. Under multi-objective scenario optimization approaches such as Tabu search, Ant colony algorithm, Particle Swarm algorithm and simulated annealing are adopted. Among those optimization approaches, performance is not desired since it involves a minimal change in objective function alters the complete performance output of the system. Thus, the need arises to develop a new SI technique. In the multi-criteria decision-making process, a clear description is not presented for type of FMS scheduling. As meta-heuristics is found to be a better option to the several objectives persisting in FMS particularly for scheduling problems, the proposed method will adopt appropriate hybrid meta-heuristic methods for various cases of scheduling problems in FMS. Also, MATLAB platform is utilized for execution as it works efficiently with high computational abilities. Also, it has built-in libraries, flexibilities to call external libraries and it is well proven in efficiently solving the optimization problems than any other tools.

CHAPTER 3
SOFT COMPUTING TECHNIQUES

3.1 INTRODUCTION

Soft computing optimization algorithms are a family of techniques designed to find the optimal solution to a problem that may involve a large number of variables or complex constraints. Soft computing techniques refer to a set of methodologies and algorithms that are designed to handle complex problems in a flexible and efficient manner. Unlike traditional computing techniques that rely on strict mathematical models and rules, soft computing techniques are inspired by biological processes and human reasoning. Soft computing techniques encompass a wide range of tools, including artificial neural networks, fuzzy logic, evolutionary computation, and swarm intelligence. These techniques are particularly useful in areas where the problem is not well-defined or the data is incomplete or uncertain. Soft computing techniques have found widespread applications in various fields, including engineering, medicine, finance, and natural language processing, to name a few. The ability of soft computing techniques to learn from data and adapt to changing conditions makes them a valuable tool for solving complex problems in the modern world.

Soft computing techniques have found a wide range of applications in scheduling and production planning due to their ability to handle complex and uncertain problems. Production scheduling involves determining the sequence of tasks, allocating resources, and assigning jobs to workers or machines to meet production targets while minimizing costs and maximizing efficiency. Soft computing techniques can be used as optimization algorithms by leveraging their ability to search through large solution spaces and identify the best possible solution based on the given criteria. An optimization algorithm is a mathematical technique that is used to find the best possible solution to a problem, given a set of constraints and objectives. Optimization algorithms are designed to search through a large space of possible solutions and identify the solution that maximizes or minimizes the objective function while satisfying the constraints.

The general working of soft computing optimization algorithms can be summarized in the following steps:

- *Problem Formulation*: The first step is to define the problem and identify the input data, output data, constraints, and objectives.
- *Data Preprocessing*: The input data is preprocessed to remove noise and inconsistencies and transform the data into a format suitable for soft computing techniques.
- *Technique Selection*: The appropriate soft computing technique is selected based on the problem requirements and characteristics.
- *Model Development*: A soft computing model is developed that can learn from the input data and generate output data based on the problem objectives and constraints.
- *Fitness Function Definition*: A fitness function (objective function) is defined that evaluates the quality of the candidate solutions generated by the soft computing algorithm.
- *Initialization*: The soft computing optimization algorithm is initialized with a set of random candidate solutions.
- *Evaluation:* Each candidate solution is evaluated using the fitness function to determine its quality.
- *Selection*: The best candidate solutions are selected based on their fitness scores.
- *Variation*: The selected candidate solutions are modified or combined to generate new candidate solutions.
- *Evaluation*: The fitness function is again used to evaluate the new candidate solutions.
- *Termination*: The optimization algorithm is terminated based on a predefined stopping criterion, such as a maximum number of iterations or a minimum level of improvement.
- *Output*: The optimized solution is outputted as the solution to the problem.

In the context of optimization, metaheuristics are high-level strategies that guide the search for optimal solutions to a given problem. Metaheuristic algorithms operate at a higher level of abstraction than traditional optimization algorithms, such as linear programming or quadratic programming, and can handle complex, non-linear, and non-convex optimization problems.

Metaheuristic algorithms are based on the idea of iteratively exploring and exploiting the search space of the problem, aiming to find the global optimum or a good approximation of it. These algorithms do not guarantee finding the optimal solution, but they are often effective at finding good solutions in a reasonable amount of time, even for very complex optimization problems. Metaheuristic algorithms are usually stochastic and iterative, meaning that they generate a set of candidate solutions and refine them over a number of iterations. The candidate solutions are evaluated using a fitness function, which measures how well each solution performs in achieving the optimization objective. The metaheuristic algorithms use this information to adjust the search strategy and generate new candidate solutions. Metaheuristic algorithms are widely used in various applications, including scheduling, routing, resource allocation, machine learning, and many others, where finding optimal or near-optimal solutions is critical for the success of the system or process.

3.1.1 Meta Heuristics inspired by Nature

Nature-inspired metaheuristics mimics the natural processes or phenomena that have evolved over millions of years to solve complex problems efficiently. These natural processes have been refined by the laws of evolution, and they have proven to be highly effective in solving complex problems in nature.

The basic principle behind nature-inspired metaheuristics is to generate a population of candidate solutions, and iteratively refine them based on a fitness function that measures how well each solution performs in achieving the optimization objective. The metaheuristic algorithms use this information to adjust the search strategy and generate new candidate solutions.

Nature-inspired metaheuristic algorithms are often stochastic and iterative, and they generate a set of candidate solutions that are iteratively refined based on a set of rules or heuristics inspired by natural phenomena. These algorithms are often used to solve optimization problems that are difficult or impossible to solve with traditional optimization algorithms.

Examples of natural phenomena that have inspired metaheuristic algorithms include:

- Evolutionary processes, such as Genetic Algorithms (GA), Differential Evolution (DE), and Genetic Programming (GP).

- Swarm behavior, such as ‘Particle Swarm Optimization’ (PSO), ‘Ant Colony Optimization’ (ACO), ‘Firefly Algorithm’ (FA) and ‘Artificial Bee Colony’ (ABC).
- Physical phenomena, such as Simulated Annealing (SA), Tabu Search (TS), Flower Pollination Algorithm (FPA), and Bat Algorithm (BA).
- Human-inspired algorithms, such as Harmony Search (HS) and Cultural Algorithms (CA).

After a thorough study and analysis of scheduling literature, the nature-inspired algorithms listed below are developed and implemented to optimize Scheduling of Flexible Manufacturing System and are described in more detail in the sections that follow:

i) GAPSOTS- A hybridization technique developed by hybridizing GA, PSO and TS
ii) HFPA- A hybridization technique developed by hybridizing classic FPA with SA
iii) HAdFA- A hybridization technique of classic FA with SA incorporated with novel adaptive strategies.

Some portions in methodologies of GAPSOTS, HFPA and HADFA are published by this researcher as part of this thesis [136]–[138]

3.2 GAPSOTS- HYBRIDIZATION OF GA, PSO AND TS

An effective multi-objective scheduling approach is applied, which is a combination of the genetic algorithm (GA), particle swarm optimization (PSO) and Tabu search (TS) algorithms (GAPSOTS algorithm). The purpose of using genetic algorithm with multiple objectives is to successfully resolve multiphase process scheduling in FMS setting. Then, PSO algorithm is applied for optimization the scheduling process and TS is used for solving combinatorial optimization issues (problems where an optimal ordering and selection of options is desired). This new approach is made by hybrid method with multiple objectives to handle the flexible job scheduling complications with manifold goals. Investigational studies have been utilized to validate the method, and a comparative analysis is done by matching the results of the recommended method to specify the compliance/ flexibility and supremacy of the present model. [136]

The *genetic algorithm* uses predetermined size of populations and considered the population issues which indicate the schedules of the machine. During every step of iteration, the non-performing or poor candidates are expelled from the population. Likewise, the schedules that are performing comparatively lesser than the expectations are detached from the population. Meanwhile, new candidates will be added to the population in GA. Thus, the fresh schedules created by means of mutations to separate schedules are substituting the detached schedules. Then the crossover operations are applied to schedule pairs existing in the population.

In the same way, *PSO algorithm* is exploited for searching the path sequence that provides an optimum answer for the problem concerned. Moreover, the concept of indigenous/local searching in the searching space is adopted for augmenting the performance of the algorithm. The searching population contain particles and every particle contained in the population relates to an individual. A particle swarm is created in arbitrary fashion at the start and the position of every individual describes a potential solution in the searching area. Two vectors namely position and velocity vectors assist to upgrade the position of every individual particle while the movement of particles take place in the searching area.

TS algorithm is fundamentally a neighbourhood approach and offers a way to clear the inflexible combinatorial problems of optimizing. It helps to get rid of "local optima" problems for such ambience. The process of transfer of present solution to the adjoining solution is termed as move. The neighbourhood/ nearby solution providing optimal solution is attained through a "move" in case of TS algorithm.

3.2.1 Framework of GAPSOTS

In the context of FMS scheduling, GA works by encoding the scheduling problem as a set of genes, and creating an initial population of potential solutions (i.e., chromosomes) randomly. Each chromosome represents a schedule that assigns specific jobs to specific machines at specific times. The fitness of each chromosome is evaluated based on its ability to minimize the objective function, such as the total completion time, makespan, or flowtime.

A hybrid approach that combines Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Tabu Search (TS) is developed to solve Flexible Manufacturing System (FMS) scheduling problems. Figure 3.1 depicts system flow process of GAPSOTS.

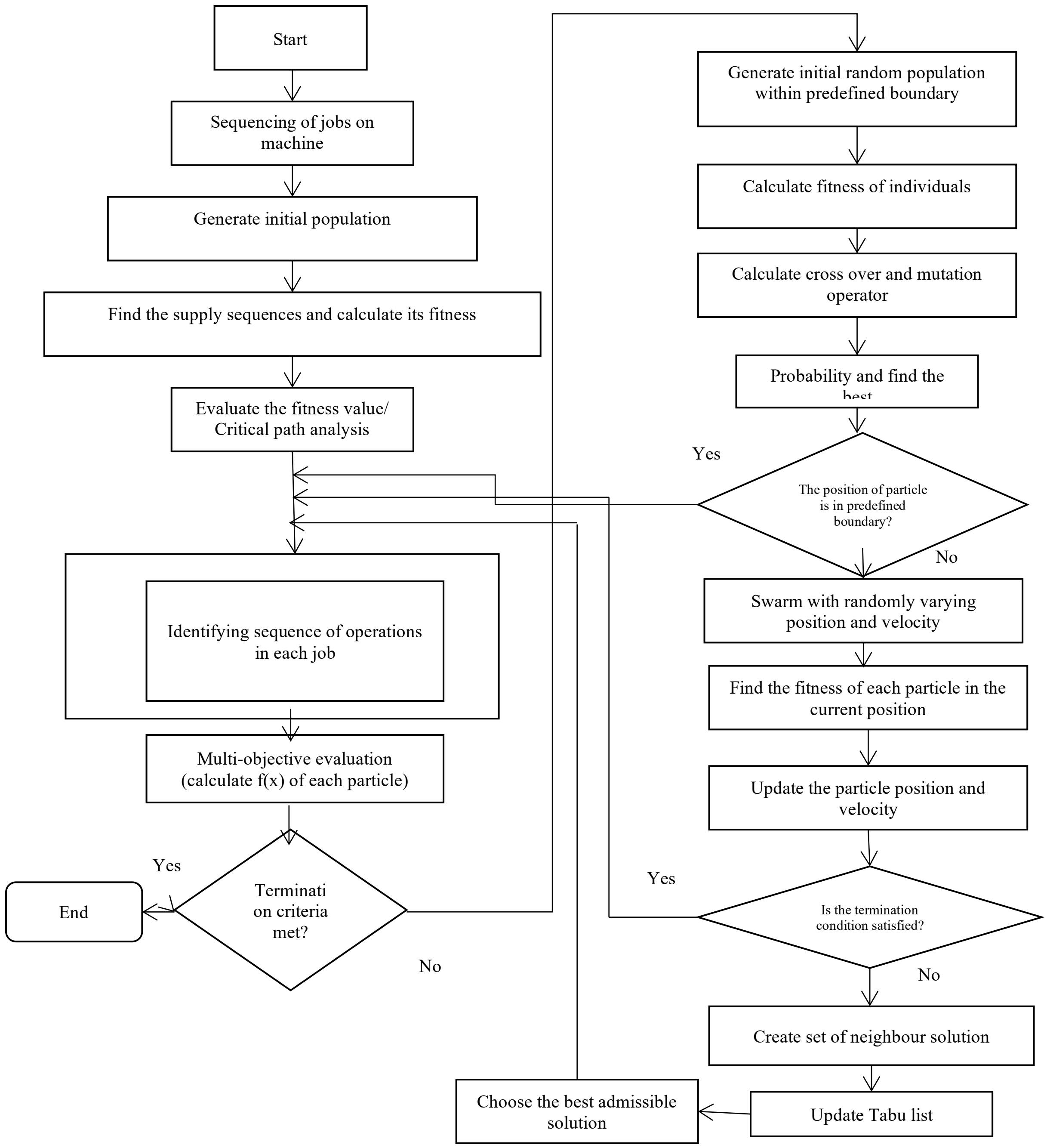


Figure 3.1 System Flow of process using GAPSOTS method

Important Aspects of GA

i) Rank Selection Method: Rank selection is a selection method used in Genetic Algorithms (GA) to select the individuals for reproduction in the next generation. In rank selection, the individuals in the population are sorted based on their fitness values, and each individual is assigned a rank based on its position in the sorted list. The best individual is assigned the highest rank, the second best individual is assigned the second-highest rank, and so on. After assigning ranks to all individuals, a probability distribution is created based on the ranks, where individuals with higher ranks have a higher probability of being selected for reproduction. Thus a mating pool with high quality parents are selected.

ii) Mutation: Swap mutation is a genetic operator used in Genetic Algorithms (GA) to introduce randomness and increase diversity in the population of candidate solutions. Swap mutation works by randomly selecting two genes (or variables) in an individual's chromosome and swapping their values.

iii) Crossover: Two-point crossover is a genetic operator used in Genetic Algorithms (GA) to generate new offspring chromosomes by combining two parent chromosomes. In two-point crossover, two crossover points are randomly selected along the length of the chromosomes, and the genes between these points are exchanged between the parents to create the offspring. This ensures that the 'child' has properties of both the parents.

Particle Swarm Optimization (PSO)

In the problem space, each individual or possible solution—referred to as a particle—flies with a velocity that is continuously changed in accordance with the collective flying experience of the particle and its companions. Each particle keeps track of its own personal best (pbest) and the swarm's best position (global best), respectively, during the search process (gbest). In order to find the best solution, each particle in the swarm interacts with the others and works to gradually advance towards the promising regions of the search space. The velocity update formula given in Equation 3.1 and 3.2 are vital aspects of PSO.

$$V_j^{t+1} = \tau\, V_j^t + C_1.rand.\left(X_j^t - pbest_j^t\right) + C_2.rand.\left(X_j^t - gbest_j^t\right) \tag{3.1}$$

$$X_j^{t+1} = X_j^t + V_j^{t+1} \tag{3.2}$$

where C_1 and C_2 are coefficients of acceleration, τ is weight, t is iteration number.

Tabu Search

The main idea behind Tabu Search is to prevent the search from revisiting recently visited solutions, which can help the algorithm to escape from local optima and explore a larger search space. A "tabu list' will be maintained which contains previous solutions, if the same solution occurs then the algorithm skips that solution and moves onto next solution. Aspiration criterion is important aspect in TS. The aspiration criterion typically involves checking the quality of the move and comparing it to the quality of the current best solution. If the move leads to a solution that is better than the current best solution, the move is allowed, even if it is currently on the tabu list. This allows the algorithm to escape local optima and potentially find better solutions.

The parameters of GAPSOTS are given in Table 3.1.

Table 3.1 Parameters of GAPSOTS

Parameters	**Value**
Population size	500
Cross over probability	0.5
Mutation probability	0.9
The total number of generations	200
Length of tabu list	10
The maximum iteration size	800
Number of particles	20

The following are the procedural steps for this hybrid approach:

1. *Problem formulation*: Define the FMS scheduling problem, including the number of machines, the number of jobs, and the processing times, and determine the objective function to be optimized.
2. *Chromosome encoding*: Encode each chromosome in the population as a schedule that assigns jobs to machines at specific times. Since FJSP has machine assignment and sequencing problem, real number encoding is used in this study.
3. *Initial population generation*: Create an initial population of chromosomes randomly. The population size can be determined based on the complexity of the problem.

4. *Fitness evaluation*: Evaluate the fitness of each chromosome based on its ability to minimize the objective function. The fitness function can be determined based on the problem formulation.
5. *Hybridization*: Combine the GA, PSO, and TS algorithms to create a hybrid algorithm. The GA is used for initial population generation and crossover, the PSO is used for particle updating and mutation, and the TS is used for local search.
6. *Selection*: Select the best individuals from the population based on their fitness to create the next generation. Rank-based selection method is used in this study.
7. *Crossover*: Exchange genetic information between two parent chromosomes to create two offspring chromosomes that inherit traits from both parents. Two point crossover is used in this study.
8. *Mutation*: Randomly change a gene or a set of genes in a chromosome to introduce diversity and prevent premature convergence. Swap mutation is used in this study
9. *Particle updating*: Update the particles in the PSO algorithm based on their position, velocity, and the best solution found so far. The update formula can be determined based on the PSO algorithm.
10. *Local search*: Apply the TS algorithm to the best solution found so far to perform a local search and improve the solution quality.
11. *New population generation*: Create a new population of chromosomes by applying selection, crossover, and mutation to the current population, and update the particles in the PSO algorithm.
12. *Convergence criteria*: Check if the termination criterion is met, such as reaching a maximum number of generations, achieving a desired fitness level, or reaching a time limit.
13. *Final solution*: Select the best chromosome from the final population as the solution to the scheduling problem.
14. *Validation*: Validate the solution by testing it on different instances of the problem and comparing it with other optimization techniques or heuristics.
15. *Parameter tuning*: Adjust the parameters of the hybrid algorithm, such as the population size, crossover rate, mutation rate, particle swarm size, and tabu tenure, to obtain better results. This step is critical for achieving optimal or near-optimal solutions.

3.3 HFPA- HYBRIDIZATION OF FPA & SA

3.3.1 Standard FPA working procedure

Flower pollination algorithm works on the basis of pollination of flowers. The pollination of a flower occurs with the help of bees, bats, wind etc. The pollination may occur between the flowers of same plant or flowers of different plants. When pollination occurs within the same plant it is called “Self-pollination or Abiotic Pollination” and when it occurs for different flowers of different plants it is called “Global Pollination or Biotic Pollination”. This ideology is used to mimic in obtaining optimal solutions in a scheduling problem.
The flowers are “solutions” and they are randomly generated at initial stage. There is a “switch operator” which decides if the pollination occurs locally or globally. Usually the “Switch Probability” ranges between 0 and 1. Table 3.2 illustrates the main components of FPA.

Table 3.2 Components of FPA for Optimization

Flower pollination	Optimization components (in FPA)
Pollinators (insects, butterflies, birds)	Moves/modification of variables
Biotic	Global search
Abiotic	Local search
Lévy flight	Step sizes (obeying a power law)
Pollen/flowers	Solution vectors
Flower constancy	Similarity in solution vectors
Evolution of flowers	Iterative evolution of solutions
Optimal flower reproduction	Optimal solution set

Yang [139] developed FPA and summarized the concept of FPA in four rules.

Rule 1 Global pollination involves biotic and cross-pollination where pollinators carry the pollen based on Lévy flights.

Rule 2 Local pollination involves abiotic and self-pollination.

Rule 3 Flower constancy can be considered as a reproduction probability that is proportional to the similarity between any two flowers.

Rule 4 Switch probability $p \in [0, 1]$ can be controlled between local pollination and global pollination due to some external factors, such as wind. Local pollination has a significant fraction p in overall pollination activities.

A random number is generated; if it is less than “p” global pollination occurs; otherwise, local pollination occurs.

The global pollination is calculated using Equation 3.3.

$$y_i^{t+1} = y_i^t + L\,(y_i^t - g^*), \tag{3.3}$$

where y_i^t is the individual flower at generation t, L is the step size obtained from the levy distribution, and g^* is the best solution among all the solutions. [138]

The local pollination is calculated using

$$y_i^{t+1} = y_i^t + \in\ (y_j^t - y_k^t) \tag{3.4}$$

where ∈ is random number from the uniform distribution [0,1], and y_j^t *and* y_k^t are different flowers. Figure 3.2 exhibits the work flow process of FPA

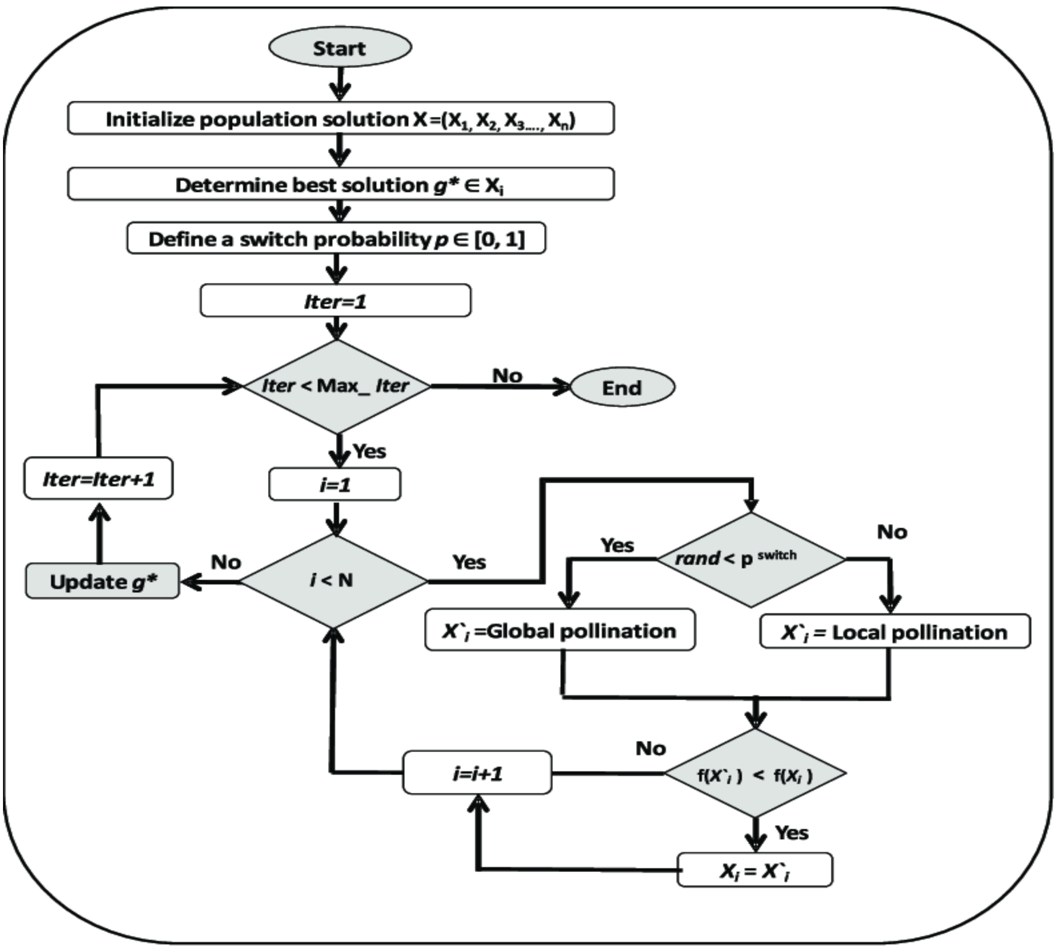


Figure 3.2 FPA work flowchart

3.3.2 Hybrid FPA

Simulated Annealing (SA) is hybridized with FPA to further improve FPA's performance. SA was chosen due to its ease of implementation and flexibility to move from the local optimum. Classic FPA cannot be used for coding since FJSP is discrete. So, in this case, DFPA suggested by Xu et al [140] is employed. The results from the DFPA are forwarded to SA for additional improvement, and the worst solution is replaced to a new best solution that SA has chosen. The best outcome found during the search process is stored and updated.

3.3.3 Encoding- Decoding of HFPA

In this study, encoding and decoding are done using the *two-string approach*. In FJSP, there are two subproblems. The operations are routed after a machine is assigned. There are two strings used: the operation routing string (ORS), which indicates the total number of operations for "n" jobs, and the machine assignment string (MAS), which indicates the total number of machines. MAS is assigned a vector that was produced at random and is equal to the number of machines. A random vector with the value n is chosen for ORS. According to the order of the activities, decoding is carried out from left to right, and related machines are distributed in accordance with MAS. Figure 3.3 depicts the discretization and encoding/decoding of HFPA. The parameters for HFPA is given in Table 3.3

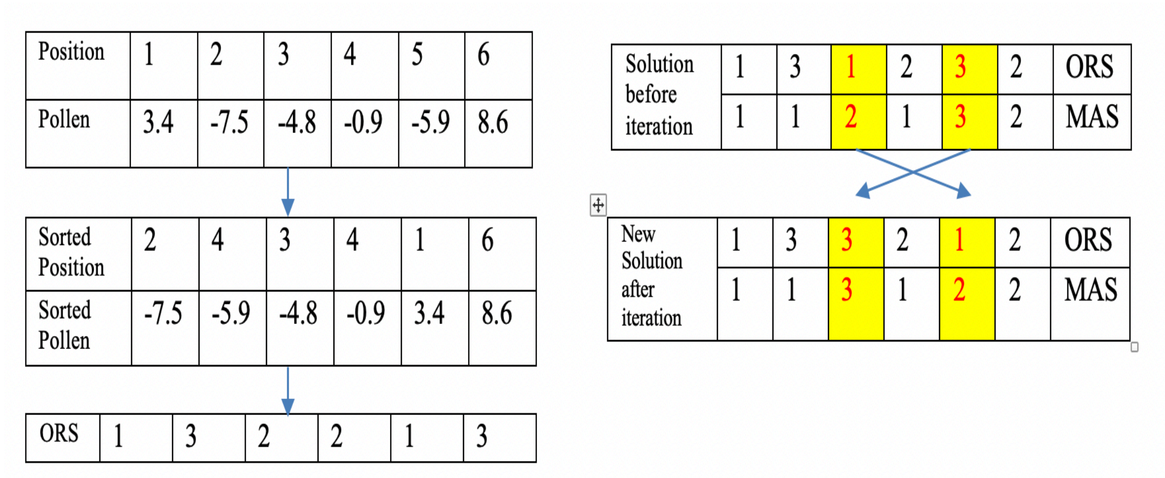


Figure 3.3 Encoding-Decoding of HFPA

Table 3.3 Parameters of HFPA

For Flower Pollination Algorithm	
Number of Flowers	50
Switch Probability	0.8
For Simulated Annealing	
Initial Temperature (T_0)	10
Temperature damping rate	0.97
Stopping Criteria	Max Iterations (1000)

3.4 HADFA (HYBRIDIZATION OF ADAPTIVE FA AND SA)

The next proposed algorithm is based upon the "bio-luminesce" property of fireflies which was proposed by Yang [141]. The firefly algorithm is a swarm-based optimization algorithm that uses a swarm of fireflies to search for the optimal solution to a given problem. Each firefly represents a candidate solution to the problem, and the swarm of fireflies explores the solution space by moving towards brighter and more attractive fireflies.

Firefly algorithm works on basic three rules and its formulas are as follows [141]

(1) All fireflies are unisexual; therefore, the fireflies may get attracted to each other regardless of their gender.

(2) If a firefly finds another firefly brighter than it, then the former will move towards it. If a firefly can't find a brighter firefly, then it roams around randomly until it finds one.

(3) The brightness of fireflies increases proportionally to the objective for a maximization problem and vice versa for the minimization problem.

(1) The attractiveness parameter β is given by Equation 3.5

$$\beta_o e^{-\gamma r^2} \tag{3.5}$$

where r is 0 and γ is the light absorption coefficient.

(2) The "movement of firefly or solution update rule" is given by Equation 3.6

$$x_{id}(t+1) = x_{id}(t) + \beta_o e^{-\gamma r^2{}_{ij}}(x_{jd}(t) - x_{id}(t) + \alpha\epsilon_i \quad (3.6)$$

Here, x_{id} and x_{jd} are the d^{th} dimension value of fireflies X_i and X_j, respectively; ϵ_i is a random value from a uniform distribution of [−0.5, 0.5]; α ∈[0, 1] is the randomization parameter; and t = 1, 2, ... is the iteration number. Since this study aims to minimize the objective function, f (X_j) is less than f (X_i), i.e., firefly X_j is better (glows brightly) than firefly X_i according to its fitness value.

The parameters β and α From Equation 3.6 plays an important role in balancing exploration and exploitation of FA. But these values are fixed in standard FA. When parameter tuning is performed for FA it is expected that the FA will perform better. Thus a novel "***Adaptive Firefly Algorithm(AdFA)***" is proposed where α- the randomization parameter changes dynamically and the whole movement formula (Equation 3.6) is replaced with new solution update formula proposed by Cheung et al [142]. With respect to the two adaptive strategies proposed , following changes are made to standard FA.

Strategy 1: The randomization α strategy proposed by Sababha et al [143] is incorporated here, which is represented by Equation 3.7.

$$\alpha(t_i) = \exp\left(1 - \left(\frac{t_{max}}{t_{max}-t_i}\right)^c\right) \quad (3.7)$$

where c (Random decaying speed) value is 5, t_{max}- Maximum Iterations, t_i- present iteration. Every iteration changes the randomness parameter from its initial value of 0.5 with Equation 3.7 which helps to hasten convergence.

Strategy 2: Cheung et al [142] proposed a "heterogenous update rule" which intensified the searching ability of fireflies. This helps the firefly to search globally for new optimal solutions. Thus Equation 3.6 is replaced by Equations 3.8a and 3.8b

$$x_{id}(t+1) = \begin{cases} (1-\tau)x_{id}(t) + \tau(x_{jd}(t) - x_{rd}(t+1), & if\ rand > 0.5 \\ \frac{G_{max}-t}{G_{max}}(1-\eta).x_{id}(t) + \eta.gbest_d(t), else & \end{cases} \quad (3.8a)$$

$$x_{rd}(t+1) = \propto (t+1) \in_i \left|x_d^{max} - x_d^{min}\right| \quad (3.8b)$$

where the "G_{max} is the maximum number of generations", "η is the gray coefficient", "rand is a random number in the range [0,1]", and "X_r is randomly generated by Equation 3.8b".

The Adaptive Firefly algorithm is further hybridized with Simulated Annealing for further fine tuning of optimal solutions obtained. The parameters of HAdFA is given in Table 3.4

Table 3.4 Parameters of HAdFA

Parameter	Value
Firefly size in the population (D)	40
The total number of generations, (k)	200
Initial attractiveness between two fireflies (β_0)	0.1
The absorption coefficient of the least intensity firefly (γ)	1
The maximum iteration size (i)	500
Randomization parameters (α)	0.5
Integer for randomness speed for α adaptiveness (c)	5
Initial temperature (T)	100
Final temperature	0.025 × Initial Temperature
Neighbourhood search structure	Insert

3.4.1 Methodological Steps for implementation of HAdFA

1. Problem Definition: Define the problem and its constraints. The constraints include the availability of resources, precedence relationships between operations, and machine setups.
2. Initialize Parameters: Define the parameters for the adaptive firefly algorithm, including the number of fireflies, the maximum number of iterations, the initial attractiveness parameter, the absorption coefficient, the step size.
3. Generate Initial Population: Generate a random initial population of fireflies.
4. Evaluate Objective Function: Evaluate the objective function for each firefly in the population. The objective function depends on the problem to be solved.
5. Sort the Fireflies: Sort the fireflies in the population based on their objective function values. The best fireflies are moved towards the global best firefly, which has the lowest objective function value.

6. Update Attractiveness Parameters: Update the attractiveness parameters for each firefly based on their distances to other fireflies. The attractiveness parameter controls the intensity of the light emitted by a firefly, which attracts other fireflies
7. Move Fireflies: Move the fireflies towards other fireflies based on their attractiveness and the distance between them. Execute new update formula Equation 3.8a and 3.8b.
8. Evaluate Objective Function: Evaluate the objective function for each firefly after moving an sort the fireflies in the population based on their objective function values.
9. Check Stopping Criteria: Check if the stopping criteria has been met. The stopping criteria is maximum number of iterations.
10. Repeat: If the stopping criteria has not been met, Execute SA
11. Initialize the initial solution for SA with a mix of best optimal results of FA and some random population.
12. Again evaluate the objective function
13. Start the annealing process by giving input parameters of annealing rate and temperature.
14. Return the Best Solution: Once the annealing process is complete, return the best solution found during the search.
15. Check Stopping Criteria if not goto step 6 repeat process until termination condition satisfied.
16. Output the final optimal results for further validation.

3.5 SUMMARY

Soft computing optimization algorithms are a family of techniques designed to find the optimal solution to a problem that may involve a large number of variables or complex constraints which helps to solve any scheduling problems with less effort. Soft computing techniques are particularly useful in solving optimization problems that involve complex constraints and objectives, where traditional optimization techniques may not be effective. In lieu of that, three new novel meta-heuristics GAPSOTS, HFPA and HAdFA are proposed for this study. The detailed working of each algorithm is explained in respective sections. Further chapters in this thesis details about the implementation of these algorithms for various scheduling problems in FMS.

CHAPTER 4
FLEXIBLE JOB SHOP SCHEDULING - A MULTI OBJECTIVE APPROACH

4.1 INTRODUCTION

Flexible job shop scheduling plays a crucial role in Flexible Manufacturing System (FMS) scheduling by helping to optimize the scheduling of production processes in highly dynamic and unpredictable manufacturing environments. Flexible job shop scheduling is a complex problem-solving technique used in manufacturing and other industries to optimize the scheduling of production processes. In a flexible job shop, there are multiple work centers, each of which is capable of performing a variety of operations. Additionally, there are several machines or resources available to perform each operation, making the scheduling process even more complex. The challenge of flexible job shop scheduling is to assign the appropriate operations to each work center and allocate the necessary resources to each operation to ensure that production runs efficiently, with minimal downtime and maximum utilization of resources. This task requires a sophisticated approach that can balance the competing demands of different production lines, resources, and deadlines, while minimizing costs and maximizing productivity. Therefore, the development of efficient scheduling algorithms and tools has become a critical research area for industrial and academic communities alike. In this context, the application of machine learning and optimization techniques can help to achieve the desired goals of flexible job shop scheduling. Flexible job shop scheduling is important in FMS scheduling because it helps to optimize the utilization of machines, resources, and labor, which is crucial for ensuring that the FMS operates efficiently and effectively. By assigning the appropriate operations to each work center and allocating the necessary resources to each operation, flexible job shop scheduling can help to reduce the time and cost of production while improving product quality.

Moreover, flexible job shop scheduling can enable an FMS to adapt quickly to changing production requirements, such as changes in demand or unexpected machine breakdowns. By providing a flexible and agile scheduling system, an FMS can respond to changing conditions quickly, helping to minimize downtime and reduce production costs.

In the context of flexible job shop scheduling, two commonly used approaches to solving the problem are the integrated method and the hierarchy method. The integrated method is a comprehensive approach that considers all the tasks involved in scheduling production processes, including task sequencing, resource allocation, and due-date assignment.

This method aims to optimize the entire production process as a whole, taking into account the interdependence and interaction between different stages of the process. The integrated method typically involves complex optimization models and algorithms, such as mathematical programming and meta-heuristics, to solve the scheduling problem. The advantage of this approach is that it can produce highly optimized schedules that take into account all the factors affecting the production process. However, the downside is that it can be computationally intensive and require significant resources to implement.

The hierarchy method, on the other hand, is a more incremental approach that breaks down the scheduling problem into smaller sub-problems and solves them sequentially. This approach involves dividing the scheduling process into different levels or stages, such as long-term planning, medium-term planning, and short-term scheduling, each of which involves different decision variables and constraints. By decomposing the problem into smaller, more manageable sub-problems, the hierarchy method can reduce the computational complexity of the scheduling process and enable faster and more practical solutions. However, the downside is that it may not always produce the most optimized schedules as it does not consider the interdependence between the different levels or stages of the production process.

Both the integrated and hierarchy methods have their advantages and disadvantages, and the choice of approach will depend on the specific requirements of the production process and the available resources. In practice, many companies may use a combination of both approaches, depending on the nature of the production method and complexity of the FJSSP.

This chapter explains the optimization of Flexible Job Shop Scheduling Problem (FJSSP) through an integrated approach by the proposed techniques HFPA and HAdFA. GAPSOTS is not implemented for this FJSSP as they are very complex in nature and it is computationally expensive when used for an integrated approach. Parts of this chapter are published as a research article by the researcher, Devi et al [137], [138]

4.2 FJSSP FRAMEWORK

The FJSSP can be articulated mathematically as a multi-objective mixed-integer linear programming (MILP) model. It typically includes decision variables, objective function, and a set of constraints that ensure that the solution satisfies the requirements of the problem. This study aims to minimize multiple objectives, including makespan, total workload, maximum machine workload, total idle time, and total tardiness of machines. The decision variables, objective functions, and limitations can be expressed as follows:

Decision Variables: Let us define the following decision variables:

- xijkm represents the start time of the operation k on the machine j at work center i.
- dijk represents the due date for the operation k at work center i.
- zij represents the completion time of the last operation at work center i on machine j.
- yi represents the total workload of work center i.
- zmax represents the maximum machine workload.
- wijkm represents the idle time of machine j at work center i between the completion of operation k and the start of operation k+1.

Objective Functions: The objective functions aim to optimize multiple objectives simultaneously. The objective functions can be formulated as follows:

- Minimize makespan:

Minimize $MS_{max} = \max(xijkm + pijk)$ (1)

- Minimize total workload:

Minimize $WL_{total} = \sum i\ yi$ (2)

- Minimize maximum machine workload:

Minimize $WL_{max} = zmax$ (3)

- Minimize total idle time:

Minimize $T_{idle} = \sum i \sum j \sum k\ wijkm$ (4)

- Minimize total tardiness of machines:

Minimize $T_{late} = \sum i \sum j \sum k \max(0, xijkm + pijk - dijk)$ (5)

Constraints: The flexible job shop scheduling problem must satisfy a set of constraints that define the relationships between different decision variables and ensure that the solution is feasible. These constraints can be formulated as follows:

- Each operation must start after its predecessor is completed. This can be expressed as:

$xijkm + pijk \leq xij'k'm$ for all $k' \neq k$ and $(i, j) \neq (i', j')$ (6)

- At a time a machine can perform only one operation. This can be expressed as:

$xijkm + pijk \leq xij'k'm + M(1-\delta ijk,ij'k')$ for all $k' \neq k$ and $(i, j) \neq (i', j')$ (7)

where M is a large positive number, and $\delta ijk,ij'k'$ is a binary variable that is 1 if operations k and k' on machines j and j' at work center i are processed simultaneously, and 0 otherwise.

- Each operation must be processed only once. This can be expressed as:

$\sum j \sum m\ xijkm = 1$ for all k and I (8)

- The start time of each operation must be greater than or equal to its release time. This can be expressed as:

$xijkm \geq rijk$ for all k, i, and j (9)

- The completion time of each operation on each machine must be tracked. This can be expressed as:

$zij \geq xijkm + pijk$ for all k, i, and j (10)

- The completion time of the last operation at each work center must be tracked. This can be expressed as:

$zij \geq zij\text{-}1k'\ m + pik'$ for all $k' \neq k$ and (i, j) (11)

- The total workload of each work center must be tracked. This can be expressed as:

$yi \geq \sum j \sum k\ pijk$ for all i (12)

- The maximum machine workload must be tracked. This can be expressed as:

$zmax \geq zij$ for all i and j (13)

4.2.1 Assumptions:

1. No setup times: The setup times required to switch between different types of machines are ignored in the problem.
2. No machine breakdowns: The problem assumes that all machines operate continuously without any breakdowns.
3. No job priorities: All jobs are assumed to have the same priority, and no preference is given to any particular job.
4. Deterministic processing times: The processing times for each operation are assumed to be known in advance and are fixed.
5. No buffer storage: The problem assumes that there is no buffer storage between machines, and each operation is processed immediately after the previous operation is completed.
6. Single-level bills of materials: The problem assumes that each job consists of a single level of bill of materials, where each operation is performed only once.
7. No capacity constraints: The problem assumes that there are no capacity constraints on the machines or work centers.

4.3 METHODOLOGIES PROPOSED

The FJSSP is attempted in this study by developing two novel metaheuristic techniques viz. HFPA and HAdFA. The detailed working methodology is already explained in Chapter 3. The objective functions to be minimized are listed in Section 4.2. The parameter settings of HFPA and HAdFA are given in Chapter 3.

In this section the implementation steps and encoding - decoding of HFPA and HAdFA to solve FJSSP are explained.

4.3.1 Implementation of HAdFA

HAdFA is the adaptive version of classical firefly Algorithm (FA). Two adaptive strategies are proposed for this study. i) In classic FA the fireflies are not updated at each step whereas in the adaptive feature the researcher proposed, the fireflies change at every updating step by a new update rule. ii) another parameter that changes dynamically is randomization parameter α. These adaptive strategies help the algorithm to strike a balance between diversification and intensification of fireflies.

Following are the steps to implement HAdFA for FJSSP:

1. *Initialize Population*: The firefly algorithm is a swarm intelligence optimization algorithm that uses a swarm of fireflies to search for the optimal solution to a given problem. Each firefly represents a candidate solution to the problem, and the swarm of fireflies explores the solution space by moving towards brighter and more attractive fireflies. The population initialization step in the firefly algorithm involves creating an initial set of fireflies, which represents the starting population of candidate solutions. The efficacy of the algorithm and the caliber of the solutions produced can be significantly influenced by the quality and diversity of the initial population.
2. *Solution Representation*: In this work, “real number encoding" is used for encoding purposes. Routing and scheduling problems are addressed using a combined strategy [144]. Each job's operations are given an integer component, while each machine's operation sequence is given a fractional component. An example is used to explain the encoding system. Table 4.1 demonstrates the matrix of three jobs and three machines along with the processing times for each machine. According to the operation's processing times in ascending order, a priority table of machines is created as shown in Table 4.2. Priority values will be assigned will be equal to the number of machines. Here 3 machines hence priority order will be for 3. And the priorities will be assigned according to the ascending order of processing time of the respective machines. When the machines have same processing times then the machine with lower number will be given the priority.

Table 4.1 Example job/machine matrix

	Position	*Operation*	*M1*	*M2*	*M3*
Job 1	1	O11	5	3	1
	2	O12	2	4	6
Job 2	3	O21	3	7	5
	4	O22	1	3	2
	5	O23	3	2	4
Job 3	6	O31	4	1	5
	7	O32	6	4	5

Table 4.2 Order of priority table

	Position	*Operation*	*Priority 1*	*Priority 2*	*Priority 3*
Job 1	1	O11	M3	M2	M1
	2	O12	M1	M2	M3
Job 2	3	O21	M1	M3	M2
	4	O22	M1	M3	M2
	5	O23	M2	M1	M3
Job 3	6	O31	M2	M1	M3
	7	O32	M2	M3	M1

Table 4.3 Stochastic illustration of firefly position

Operations	*O11*	*O12*	*O21*	*O22*	*O23*	*O31*	*O32*
Firefly Locus	2.564	3.321	1.035	2.245	1.234	2.987	3.032
Level of Priority	2	3	1	2	1	2	3
Machine Number	M2	M3	M1	M3	M2	M1	M1

Table 4.3 depicts the illustration of fireflies when assigned the random integers with fractions. The maximum number of integers randomized will be equal to the number of machines plus 1 and the least value should be 1. According to the matrix given in Table 4.3, O11 has 2.564 hence 2 will be the priority order and the corresponding machine will be assigned to it. Same way all operations will be alloted machines. Following the allocation, each machine's tasks are carried out in ascending order of fractional values. This was both machine assignment and its sequencing are done concurrently.

3. *Firefly Assessment*: The firefly assessment is done through the HAdFA implementation. The minimization of objective functions given in Section 4.2 is measured. Adaptive strategies are implemented. Update the solutions according to the new update rule. Evaluate the best result and quit if termination criteria are achieved.

4.3.2 Implementation of HFPA

The Flower Pollination Algorithm (FPA) is inspired by the process of pollination in flowers. The FPA algorithm simulates this process by representing the solutions to an optimization problem as flowers, and the search for the optimal solution as a process of pollination. The algorithm generates new solutions by combining the features of two or more solutions, similar to the way that pollinators carry pollen from one flower to another. The Lévy flight operation used in the FPA algorithm simulates the movement of pollinators between flowers, and the local search algorithm Simulated Annealing (SA) represents the process of refining the solutions over time.

The HFPA algorithm is a powerful optimization technique that leverages the natural processes of pollination to efficiently search for optimal solutions to complex problems. By mimicking the behaviour of pollinators and flowers and hybridizing FPA with SA, the proposed HFPA algorithm is able to generate high-quality solutions and quickly converge to an optimal solution.

Steps to implement HFPA is similar to HAdFA. First initialize population, input the parameters, Evaluate Objective Function by HFPA, if satisfied quit else continue the iterations. The discrete FPA is adapted for this study. The detailed steps can be further studied from Xu et al[140].

Methodological steps of HFPA

- Initialize a population of solutions randomly.
- Generate new solutions by combining features of two or more solutions using Lévy flight operation.
- Improve the quality of the solutions through a local search (SA).
- Select the best solutions for the next generation.
- Repeat steps 2-4 till termination conditions are met.

4.4 INPUT TEST INSTANCES

The FJSSP is solved for objectives to minimize makespan, maximum machine workload, total workload, total machine idle time and tardiness.

The following five benchmark test instances are used to validate the proposed HFPA and HAdFA. All the data sets are easily accessible from OR library. The comprehensive data instances for this problem can be found in https://d-nb.info/1023241773/34

1. *Kacem data set*. [145] This is the most solved data set in FJSSP for decades. This data set consists of 5 problems with various machines and jobs and operations. The data of Kacem is given in Appendix Table A 1.1- Table A 1.5
2. *Dauzère–Pérès data set [DP data set]* [146]. This data set has 18 problems in total. Each problem has varying jobs (10-20), machines (5-10), operations (15-25).
3. *Brandimarte's Data set* [*BR Data Set*] [147] This benchmark problem has 10 problems with machines varying from 15-20, jobs varying from 10-20 with 240 operations in total. It is the standard benchmark problem for FJSSP.
4. *Du Test Instances and Rajkumar Instance* [148], [149] This data set consists of 3 problems with 8 jobs to 12 machines for 30 operations.

4.5 PROPOSED ALGORITHMS VALIDATION AND DISCUSSIONS

Using Matlab R2019b, the proposed HAdFA and HFPA have been developed and validated. To show that our algorithms are more effective than other metaheuristics reported in the literature, 50 simulation runs were done to ensure the program's stability. In the sections that follow, the performance assessments of the algorithms for benchmark examples are covered in more detail. This section describes the performance comparison of HAdFA with HFPA and other metaheuristics that have been used to explicate the benchmark instances in the literature.

4.5.1 Kacem Data sets (KA data set)

Performance Comparisons of HFPA with HAdFA

Table 4.4 shows the performance comparisons of HFPA and HAdFA. The Column 1 represents the problem type, with 'n' jobs and 'm' machines and number of operations 'o'. The objectives that have been solved are MS_{max}, WL_{max}, WL_{total}, T_{late}, and T_{idle}. In every problem, HAdFA performs better than HFPA, particularly for large problem cases. HAdFA performs better than HFPA owing to the adaptive parameter technique it uses. Although the processing

times of the two approaches hardly differ, real-time complex issues can benefit more from those few seconds. For each Kacem instance, the best possible solution is depicted as Gantt charts in Figures 4.1–4.5. The Gantt chart's ordinate indicates the number of machines, while the abscissa indicates for time. In order to make the Gantt chart easier to interpret, each job is given a distinct color. Below the relevant job block, processing times for that job is given.
J 1,1 denotes Job 1, with Operation 1, J 2,3 denotes Job 2, with Operation 3, and so forth. Every task represented on a Gantt chart is subject to the same rule. Since HADFA performs better the gantt charts are shown for HAdFA technique only.
Note: The entire performance comparisons for this data set are published as a research article by the researcher, Devi et al [138]

Table 4.4 Comparison of HAdFA and HFPA for Performance Metrics

Problem	HAdFA						HFPA					
n x m x o	MS_{max}	WL_{max}	WL_{total}	T_{idle}	T_{late}	T(s)	MS_{max}	WL_{max}	WL_{total}	T_{idle}	T_{late}	T(s)
4 × 5 × 12	10	8	32	11	8	1.1	11	10	32	9	6	1.4
8 × 8 × 27	14	12	74	22	21	1.1	14	12	77	22	19	1.1
10 × 7 × 29	11	11	61	10	41	2	11	10	62	7	43	2.5
10 × 10 × 30	7	6	41	11	20	3	7	6	42	12	30	3
15 × 10 × 57	11	11	91	20	38	13	11	10	93	25	38	10

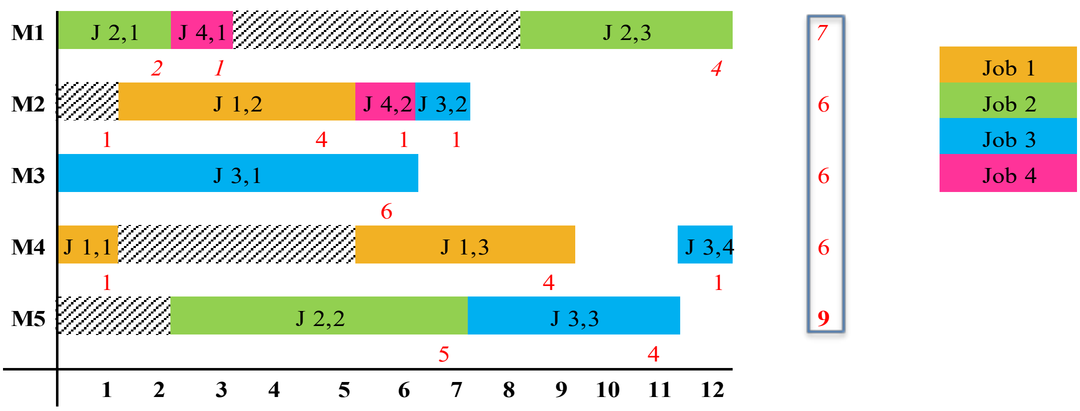


Figure 4.1 HAdFA KA1 (4 × 5 × 12) MS_{max} = 12 WL_{max} = 9 WL_{total} = 34

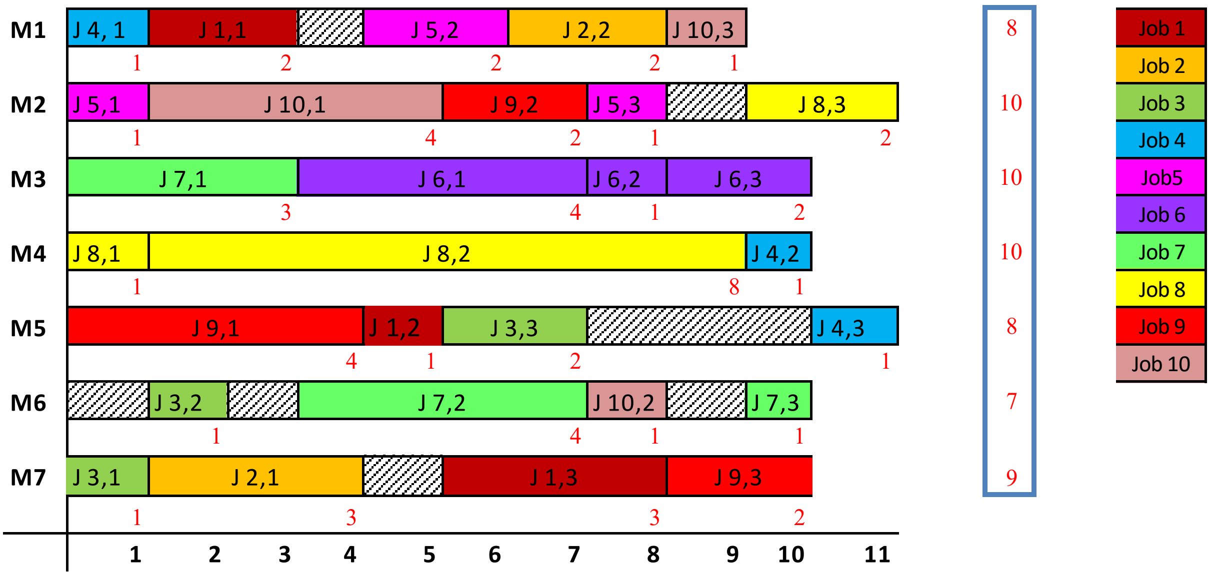


Figure 4.2 HAdFA- KA2- (10 × 7 × 29) MS_{max} = 11 WL_{max} = 10 WL_{total} = 62

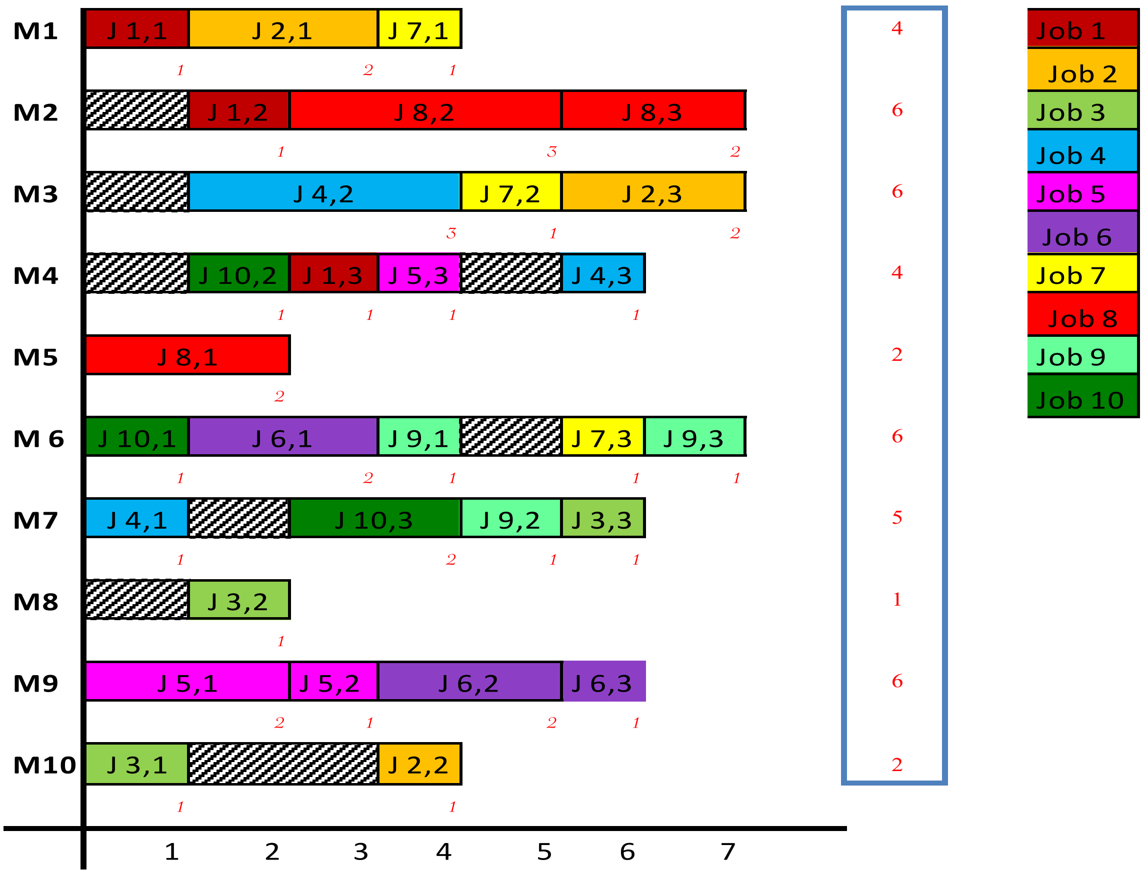


Figure 4.3 HAdFA -KA-3 (10× 10× 30) MS_{max} = 7 WL_{max} = 6 WL_{total} = 42

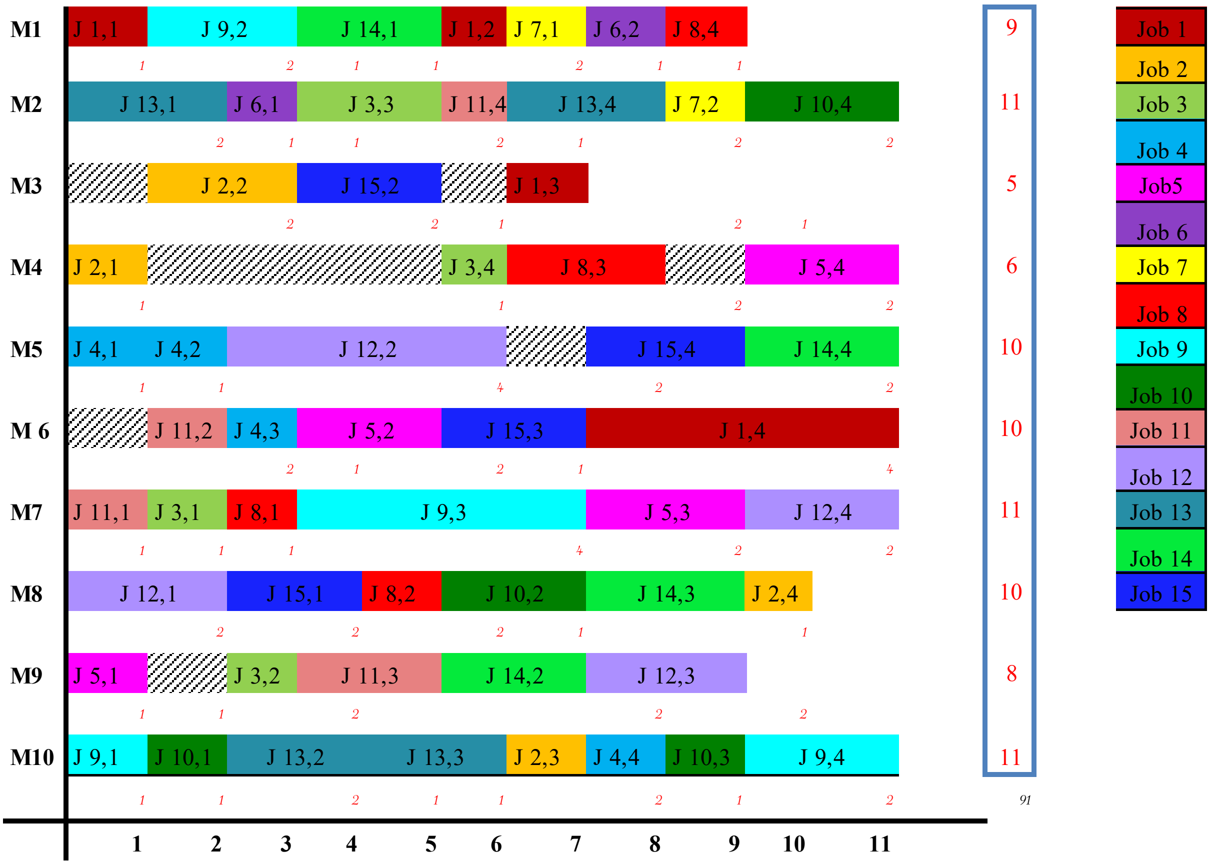


Figure 4.4. HAdFA- KA-4 (15 × 10 × 57) MS_{max} = 11 WL_{max} = 11 WL_{total} = 91

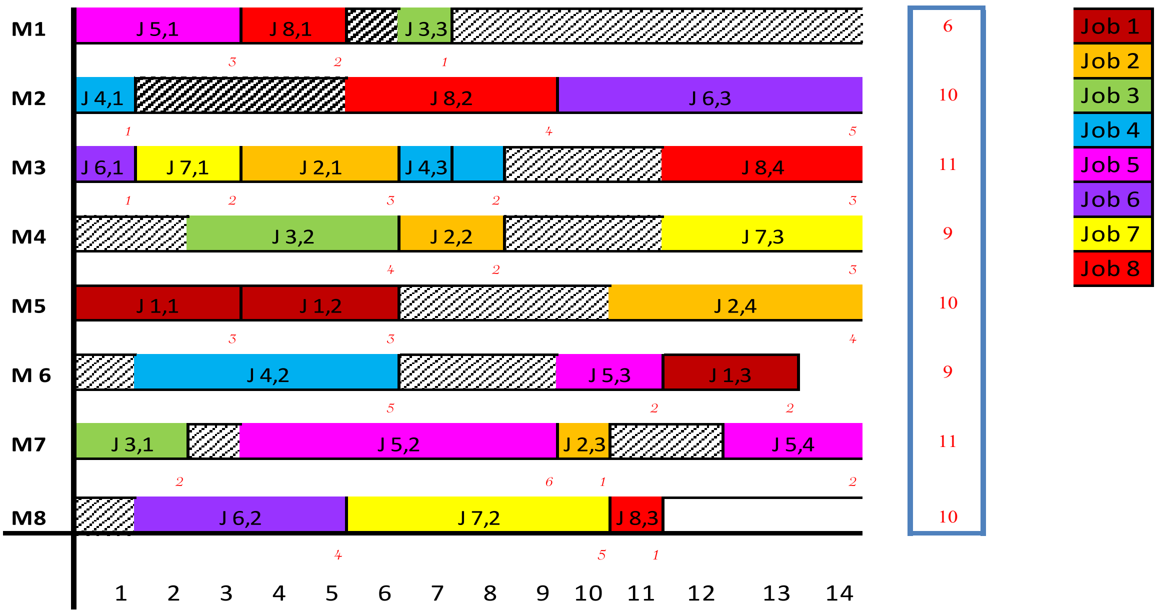


Figure 4.5 HAdFA-KA-5 (8 × 8× 27) MS_{max} = 14 WL_{max} = 11 WL_{total} = 76

Performance Comparison of Proposed Algorithms with Other Meta-Heuristics [138]

In this section the proposed algorithms are measured up against existing algorithms found in the literature. In view of the fact that there are not any previous studies comparing the T_{late} and T_{idle} values, to the best of researcher's knowledge, Table 4.5 presents the comparison of HAdFA and HFPA with existing algorithms for MS_{max}, WL_{max}, WL_{total} only. Results of HAdFA and HFPA are compared with DE [150], MOPSO [151], BEG NSGA II [152], PSO+RRHC [153], ADCSO [154], and IH PSO [155]. NA indicates the nonavailability of data for that particular problem. The comparison has been made with similar multi-objective problems and the most recent technique. In the 4 × 5 problem, two new solutions of MS_{max} = 10, WL_{max} = 8, and WL_{total} = 32, and MS_{max} = 12, WL_{max} = 9, and WL_{total} = 34 were found. The Gantt chart for the new Pareto is shown in Fig 2. Similarly, in the 8 × 8 problem, two new solutions, MS_{max} = 16, WL_{max} = 13, and WL_{total} = 74 and MS_{max} = 14, WL_{max} = 11, and WL_{total} = 76 were found. For the 10 × 7 problem, MS_{max} = 12, WL_{max} = 11, and WL_{total} = 62 was found. The newly found solutions are highlighted. Table 4.5 also illustrates that HAdFA provides new Pareto optimal solutions in addition to giving best optimal solutions. In Table 4.5, the last column, T, displays the computational time in 's'. The low computational power aids in running the program multiple times by changing the parameters as the researcher wants. This advantage will be helpful when solving for real time case studies.

Table 4.5 Experimental Validation for Kacem Data Set with other Metaheuristics.

	DE			MOPSO			BEG NSGA			PSO+RRHC			ADCSO			IH PSO			HAdFA			HFPA		
	F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3	F1	F2	F3
KA1	11	8	32	11	10	32	11	10	32				12	8	32	11	10	33	**10**	**8**	**32**	11	10	32
	11	10	31	13	7	33	11	9	34				11	10	32				11	10	32	12	8	32
				12	8	32	12	8	32										11	8	31			
							13	7	33										**12**	**9**	**34**			
KA2	11	12	62										11	10	62				11	12	62	11	10	62
	11	13	61										11	11	61				11	11	61	11	11	61
	12	11	64										12	12	60				11	10	62	12	12	60
																			12	11	62			
KA3	8	5	41	8	7	41	7	5	43	7	5	43	7	6	42	7	6	42	7	6	42	7	5	43
	7	6	42	8	5	42	7	6	42	7	6	42	7	5	43	8	7	41	7	5	43	7	6	42
	7	5	45	7	5	43	8	5	42	8	5	42	8	5	42				8	5	41	8	7	41
				7	6	42	8	7	41	8	7	41							8	5	42			
KA4	11	11	91	11	11	91	11	11	91	11	11	91	12	11	91				11	11	91	11	11	91
	12	10	95	11	10	93	12	10	93	14	10	93	11	11	92				11	10	95	11	10	93
KA5	14	13	75	16	13	73	14	12	77	14	12	77	16	13	73	14	12	77	14	12	77	14	12	77
	15	14	74	14	12	77	15	12	75	15	12	75	15	12	75	15	12	75	**14**	**11**	**76**	15	12	75
				16	11	77	16	11	77	16	11	77	14	12	77				**16**	**13**	**74**	16	13	73
				15	12	75	16	13	73	16	13	73							16	11	77			

F1=MS_{max}; F2= WL_{max}; F3= WL_{total}

4.5.2 Dauzère–Pérès data set [DP data set]

Dauzère–Pérès [146] developed this test instance to test his algorithm. This data set known as 'DP data set' is a standard benchmark problem often used by researchers to validate their developed algorithm. Table 4.6 shows the performance assessment of the computational results of the DP data of HFPA and HAdFA with the hybrid GA (hGA) proposed by Gao et al. [156] Column T in Table 4.6 indicates the average computational time in 's'; the last column shows the *Relative Deviation Measure* obtained by comparing the best makespan of hGA, HFPA, and HAdFA to the makespan of another algorithm. The best makespan obtained by the three algorithms is indicated in bold letters. The results of this DP data set are published as research article by the researcher, Devi et al[138].

Table 4.6 Experimental Results for DP data set

Problem	n x m	Flex	(LB,UB)	hGA			T	HFPA			T	HAdFA			T	PI
				F1	F2	F3	seconds	F1	F2	F3	seconds	F1	F2	F3	seconds	%
01a	10 x 5	1.13	(2505, 2530)	2518	2505	11 137	102.71	**2505**	2505	11 137	20.5	**2505**	2505	11 137	19.8	0.51%
02a	10 x 5	1.69	(2228, 2244)	2231	2231	11 137	140.98	2231	2231	11 137	27.1	**2230**	2230	11 137	12.55	0.04%
03a	10 x 5	2.56	(2228, 2235)	**2229**	2229	11 137	106.53	2229	2229	11 137	15.32	2229	2229	11 137	12.45	0.00%
04a	10 x 5	1.13	(2503, 2565)	2515	2503	11 085	95.93	2507	2503	11 085	34.23	**2503**	2503	11 074	31.03	0.47%
05a	10 x 5	1.69	(2189, 2229)	2217	2217	11 045	143.95	2219	2212	10 986	30.05	**2215**	2212	10 981	31.85	0.09%
06a	10 x 5	2.56	(2162, 2216)	**2196**	2196	10 962	111.83	2197	2194	10 893	25.2	**2196**	2193	10 891	20	0.00%
07a	15 x 8	1.24	(2187, 2408)	2307	2287	16 485	356.32	2298	2190	16 485	40.34	**2282**	2187	16 485	36.23	1.10%
08a	15 x 8	2.42	(2061, 2093)	2073	2070	16 485	330.08	**2069**	2069	16 485	39.87	2070	2069	16 485	45.32	0.19%
09a	15 x 8	4.03	(2061, 2074)	2066	2065	16 485	327.49	2066	2066	16 485	12. 05	**2065**	2065	16 485	10	0.00%
10a	15 x 8	1.24	(2178, 2362)	2315	2263	16 532	345.19	2307	2293	16 487	21.9	**2291**	2257	16 494	18.08	1.05%
11a	15 x 8	2.42	(2017, 2078)	2071	2069	16 418	360.45	2069	2063	16 228	61.23	**2064**	2058	16 230	21.33	0.34%
12a	15 x 8	4.03	(1969, 2047)	2030	2030	16 172	329.71	2030	2024	16 065	31	2030	2022	16 065	29.05	0.00%
13a	20 x 10	1.34	(2161, 2302)	2257	2254	21 610	462.85	2257	2244	21 610	22.35	2257	2233	21 610	22	0.00%
14a	20 x 10	2.99	(2161, 2183)	2167	2164	21 610	587.13	2167	2167	21 610	31.05	2167	2164	21 610	28.85	0.00%
15a	20 x 10	5.02	(2161, 2171)	2165	2165	21 610	669.92	2166	2165	21 610	10.05	2165	2165	21 610	19.35	0.00%
16a	20 x 10	1.34	(2148, 2301)	2256	2242	21 593	452.41	2256	2251	21 534	24.89	**2255**	2243	21 504	18.78	0.04%
17a	20 x 10	2.99	(2088, 2169)	2140	2138	21 307	616.34	2140	2138	21 114	21.3	**2138**	2138	21 105	9	0.09%
18a	20 x 10	5.02	(2057, 2139)	2127	2127	21 204	667.01	2127	2127	21 009	15.6	2127	2125	21 006	18.9	0.00%

F1=MS_{max}; F2= WL_{max}; F3= WL_{total}

Table 4.6 shows that our proposed HFPA and HAdFA exhibit superior performance to hGA. New MSmax values are found for almost all problems by HAdFA. Significantly, HAdFA achieved well in, more than 60% of the total problems. For Problem 8a, HFPA afforded a better MSmax than hGA and HAdFA. The computational time for HFPA and HAdFA was drastically reduced compared to hGA. Although the computational time depends on the operating system, the speed of our proposed techniques is still superior to hGA. Our proposed techniques take a few seconds to run a problem. The test problems were run on Intel Core i7 2.3-GHz CPU by Matlab 2019 and above. All the problems were simulated 25 times to assess the constancy of the code, and satisfactory results were obtained. Comparing HFPA and HAdFA results for DP data, the computational time of HAdFA is almost similar to that of HFPA. HAdFA affords the best makespan in the majority of cases. Hence, the HAdFA performs better than the HFPA for DP data. The computational results validate that both HFPA and HAdFA perform better than hGA in terms of their running time and objective results as well

Convergence Analysis

To better illustrate the performance superiority of HAdFA over HFPA, a convergence pace curve is shown in Figure 4.6 for ***DP data instance-10a***. The X axis is CPU time in seconds and Y axis is Makespan value. The makespan obtained by HFPA is 2307 in 22 seconds and HAdFA obtained makespan value of 2291 in 18 seconds. Figure 4.6 clearly depicts the faster convergence of HAdFA in few seconds than HFPA to achieve the lower makespan value. This analysis holds for maximum test problems. However, for few problems HAdFA performed slow in comparison to HFPA, especially when number of operations are more. But obtaining best makespan compensates for the little higher computation time.

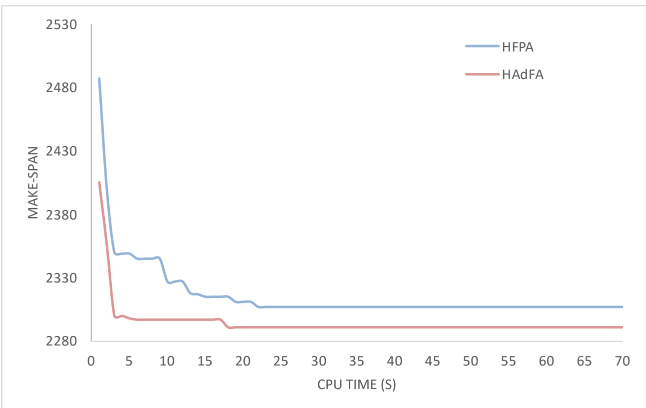


Figure 4.6. Convergence Graph for DP data-10a instance of the HFPA and HAdFA.

4.5.3 Brandimarte's Test instances (BR data set)

The input data for BR data set was originally developed by Brandimarte [147]. This became a benchmark problem set in later years to validate any newly developed algorithms for FJSSP. The BR data set consists of 10 problems with various jobs size and number of machines. Total of 240 operations are performed in this data. Combined objective function of minimization of makespan, maximum workload, total workload is also attempted for this data set. Weighted Sum approach is used for this. Equation 14 gives COF formula.

$$\text{Minimize} \quad COF = \omega_1 \text{ x } MS_{max} + \omega_2 \text{ x } WL_{max} + \omega_3 \text{ x } WL_{total} \tag{14}$$

Subject to:

$$\omega_1 + \omega_2 + \omega_3 = 1 \quad 0 \leq \omega_1, \omega_2, \omega_3 \leq 1 \tag{15}$$

where COF is the combined objective function. $\omega_1, \omega_2, \omega_3$ are weights given for the three objectives. The problem determines the weight value; if an objective requires more consideration, a higher weight value is assigned; otherwise, a lower value is given. The weighted sum approach has the benefit of allowing the user to adjust the weights of the objectives used in accordance with the problem requirement. Weights assigned are ω_1=0.7 ω_2=0.15 ω_3=0.15 for this study. As its already proved that HAdFA gives better results than HFPA from previous analysis (Section 4.5.1, 4.5.2) the BR data study is focussed to emphasize the significance of the performance of adaptative parameters employed in FA, the results of BR data problems are compared with Discrete FA (DFA) proposed by Lunardi et al [157] , Adaptive FA and HAdFA. Table 4.7 displays the results of DFA, AdFA and HAdFA. The last column in the table indicates the percentage improvement (PI) of HAdFA with AdFA.

Table 4.7 Results of DFA, AdFA and HAdFA with PI

Problem	Technique	MS_{max}	WL_{max}	WL_{total}	COF	PI for COF
MK01	AdFA	40	36	150	45	2.22%
	HAdFA	40	36	150	44	
MK02	AdFA	26	25	136	32.85	**5.79%**
	HAdFA	26	21	128	31.05	
MK03	AdFA	204	159	831	212.05	0.56%
	HAdFA	204	146	822	210.85	
MK04	AdFA	60	63	335	74.05	0.54%
	HAdFA	54	53	324	73.65	
MK05	AdFA	172	171	669	198.45	1.58%
	HAdFA	170	170	658	195.35	

Table 4.7 contd

MK06	AdFA	55	62	330	79	2.27%
	HAdFA	53	52	332	77.2	
MK07	AdFA	136	139	676	159.35	1.47%
	HAdFA	138	136	648	157	
MK08	AdFA	523	520	2478	619.05	0.78%
	HAdFA	523	506	2410	614.25	
MK09	AdFA	299	311	2251	406	1.05%
	HAdFA	300	299	2249	401.75	
MK10	AdFA	211	219	1896	305.35	1.83%
	HAdFA	**201**	212	1872	299.85	

From Table 4.7, it can be observed that best makespan values are obtained for large scale problems especially MK10 has new makespan value of 201. Though there is not much difference in makespan value between DFA with AdFA and HAdFA for small and mid-size problems, large scale problems show considerable improvement. Also, the COF shows a significant enhancement in all cases. This demonstrates the importance of adaptive parameters and local search when introduced to classic FA. It is also observed that there is a significant improvement of 2.22%, 5.79%, 1.58%, 2.27%, and 1.47% in COF for MK 01, MK02, MK05, MK06, and MK07, respectively. Though MK03, MK04, MK08 and MK09 show a little improvement in optimal values, this slight difference will significantly impact the computational time.

A Wilcoxon signed rank test is adopted to analyze the significant differences in the results for each test instance. It's a non-parametric analysis used for performance comparison of two metaheuristics. A confidence level of $\alpha = 0.05$ is taken. The analysis was done for average makespan of 10 Brandimarte instances for 25 runs. The results show that the HAdFA performs better than Adaptive FA for 8 instances with p-value < 0.0001 except MK01, MK03 and MK08 for which p-value is '0' that indicates there is no significant difference in both algorithms. For MK07 and MK09 AdFA performs better than HAdFA.

Performance Comparison of HAdFA with other Techniques [137]

Next, the proposed HAdFA is compared with other recent techniques taken from literature, and the comparison is shown in Table 4.8. Very few studies have been done to solve COF. The proposed HAdFA is compared with Search Method (SM) by Xing et al. [158], HTSA by Li et al. [159] , MODE by Balaraju et al. [150], BEG-NSGA-II by Deng et al. [152], and

ADCSO by Jiang et al. [154]. The last column gives percentage of algorithm improvement. (AI)

Table 4.8 Comparison of Performance metrics for Brandimarte problems

Problem	HAdFA				ADCSO				BEG-NSGA-II				SM				HTSA				MODE				AI
	COF	Wt	Wm	Cm	COF	Wt	Wm	Cm	COF	Wt	Wm	Cm	COF	Wt	Wm	Cm	COF	Wt	Wm	Cm	COF	Wt	Wm	Cm	
MK01	44	143	31	**40**	45.75	167	36	40	NA	154	36	42	48	162	42	42	45.75	167	36	**40**	NA	154	36	**40**	0%
MK02	31.05	128	21	**26**	32.8	143	27	27	NA	140	26	**26**	34.35	155	28	28	32.35	151	26	**26**	NA	141	26	**26**	0%
MK03	210.85	822	146	**204**	239.5	914	204	204	NA	847	133	**204**	236.4	852	204	204	236.4	852	204	204	NA	845	133	204	0%
MK04	73.65	324	53	**54**	77	362	62	62	NA	339	55	60	82.05	353	67	68	76.25	366	61	61	NA	332	55	62	**11%**
MK05	195.35	658	170	170	198.6	685	173	173	NA	674	173	173	203.25	702	177	177	197.75	687	172	172	NA	672	172	172	**1%**
MK06	77.2	332	52	53	81.65	397	60	66	NA	330	54	60	91.6	67	431	75	81.2	398	62	65	NA	346	55	60	**13%**
MK07	157	648	136	138	170.75	698	143	143	NA	657	138	139	178.35	717	150	150	167.755	695	140	140	NA	658	138	139	**1%**
MK08	614.25	2410	506	523	623.05	2524	523	523	NA	2484	515	523	623.05	2524	523	523	623.05	2524	523	523	NA	2480	515	523	0%
MK09	401.75	2249	299	300	408.45	2293	300	311	NA	2265	299	310	412.35	2374	299	311	407.85	2294	301	310	NA	2260	299	310	**3%**
MK10	299.85	1872	212	215	308.5	1943	209	225	NA	1864	211	224	314.2	1989	221	227	305.35	2053	210	214	NA	1890	214	229	**-0.46%**

where Cm- MS_{max}; Wm- WL_{max}; Wt- WL_{total}

The results shown in Table 4.8 indicate that our technique achieves the best output for MK01–MK03 and MK10, and it outperforms other approaches, especially in mid-scale problems. A maximum improvement of 13% can be seen for MK06. All the problems have been run 25 times. From Table 4.8, it can be inferred that proposed algorithm works effectively for both Single-objective optimization and multi-objective optimization. There is a substantial improvement in both computational times and the optimal results. As the actual optimal schedule for Brandimarte instances is very challenging to be represented as Gantt charts, two sample Gantt charts are shown in Figure 4.7 and 4.8, and these charts are obtained from HAdFA for MK01 and MK06 for make-span. The hatched lines indicate idle time.

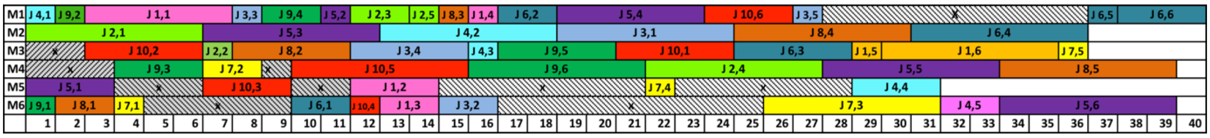


Figure 4.7 Gantt chart of problem MK01 for make-span (MSmax = 40)

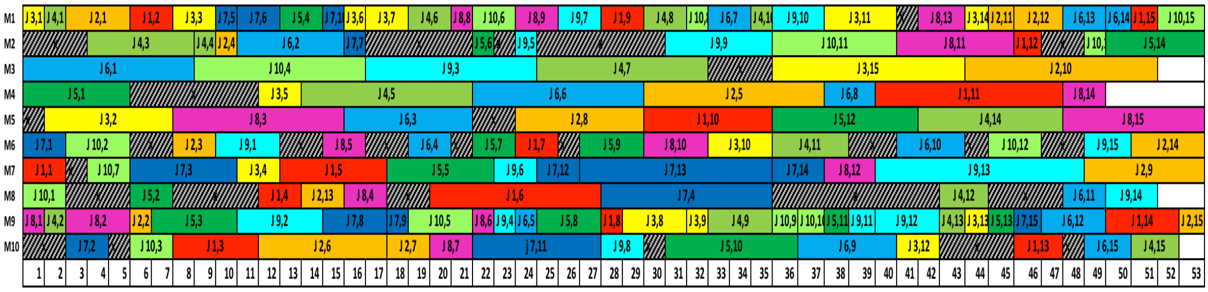


Figure 4.8 Gantt chart of problem MK06 for make-span (MSmax = 53)

4.5.3 Du Test Instances and Rajkumar Instance

In this section three problem instances viz problems with 8jobs-5machines, 12jobs–5machines, and 8jobs–8machines are utilized to study the efficacy of Adaptive FA and HAdFA. The problem sets are originally developed by Du et al., [148] and Rajkumar et al. [149]. For performance comparison, the maximum and average results from experiment data of 20 distinct runs were gathered.

The computational results obtained by AdFA and HAdFA are given in Table 4.9. Table 4.9 also shows the COF obtained for different weights and performance comparison with other algorithms, including GA by Du et al. [148], GRASP technique by Rajkumar et al. [149], Discrete FA (DFA), and Hybrid Discrete FA (HDFA) by Karthikeyan et al.[160]. The '-' indicates that the data is unavailable. The time taken for computation is given in seconds. It is validated that the proposed AdFA and HAdFA perform superiorly to the other algorithms in reference to the results shown in Table 4.9. The computational time has been reduced drastically in comparison to other algorithms. The computational time of AdFA and HAdFA has significantly less difference. There is a significant improvement in COF values. Figure 4.9, 4.10 and 4.11 depicts the Gantt chart of solutions. J (11), J (12), and so on represent the job number and operation in Gantt charts. Hatched lines show the machine's idle time.

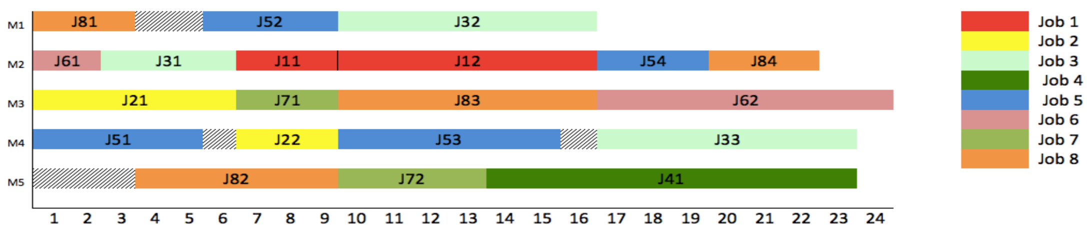


Figure 4.9 Gantt chart of problem 8×5 with 20 operations

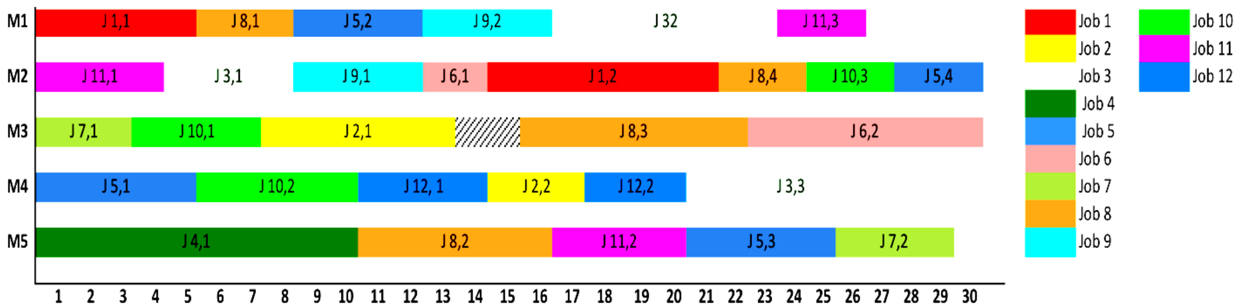


Figure 4.10 Gantt chart of problem 12×5 with 30 operations

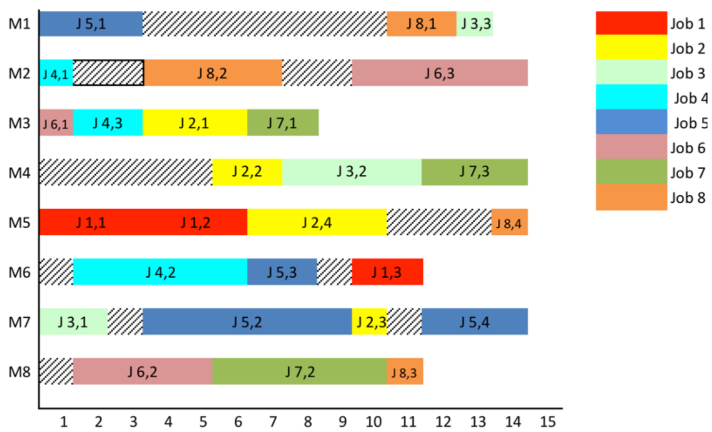


Figure 4.11 Gantt chart of problem 8×8 with 27 operations

Table 4.9 Results of performance metrics for Du and Rajkumar Test instances

Problem			8x5x20	12x5x30	8x8x27
GA	MS_{max}		27	33	-
	WL_{max}		27	33	-
	WL_{total}		109	145	-
	COF	F(0.5-0.3-0.2)	43.4	55.4	-
		F(0.3-0.2-0.5)	68	89	-
		F(0.2-0.3-0.5)	51.6	66.6	-
	Time(s)		-	-	-
GRASP	MS_{max}		24	33	16
	WL_{max}		24	33	13
	WL_{total}		101	138	73
	COF	F(0.5-0.3-0.2)	39.4	54	26.5
		F(0.3-0.2-0.5)	62.5	85.5	43.9
		F(0.2-0.3-0.5)	47.1	64.5	31.6
	Time(s)		NA	NA	-
DFA	MS_{max}		28	34	16
	WL_{max}		25	32	13
	WL_{total}		102	139	73
	COF	F(0.5-0.3-0.2)	41.9	54.4	26.5
		F(0.3-0.2-0.5)	64.4	86.1	43.9
		F(0.2-0.3-0.5)	49.7	64.5	31.6
	Time(s)		0.18	0.37	0.41
HDFA	MS_{max}		24	31	15
	WL_{max}		24	30	12
	WL_{total}		101	140	75
	COF	F(0.5-0.3-0.2)	39.4	52.5	26.1
		F(0.3-0.2-0.5)	62.5	85.3	44.4
		F(0.2-0.3-0.5)	47.1	63.2	31.5
	Time(s)		0.45	0.93	0.85
AdFA	MS_{max}		24	31	15
	WL_{max}		24	30	12
	WL_{total}		100	140	75
	COF	F(0.5-0.3-0.2)	39.4	52.8	26.5
		F(0.3-0.2-0.5)	62.5	85.1	44
		F(0.2-0.3-0.5)	47	64.1	31.5
	Time(s)		0.28	0.4	0.45
HAdFA	MS_{max}		22	30	14
	WL_{max}		18	30	12
	WL_{total}		97	140	75
	COF	F(0.5-0.3-0.2)	39.4	51.5	25.85
		F(0.3-0.2-0.5)	60	85	43.2
		F(0.2-0.3-0.5)	45	62.9	31.5
	Time(s)		0.25	0.39	0.4

4.6 SUMMARY

Flexible job shop scheduling problem (FJSSP) with an aim to solve multi-objectives of minimizing makespan, maximum workload, total workload, total idle time, total tardiness is performed by the proposed HFPA and HAdFA. Four benchmark instances viz; Kacem intances, Brandimarte instances, Dauzère–Pérès, Du and Rajkumar instances are utilized to validate the proposed algorithms efficiency. Total of 36 problems with varying job and machine size are evaluated. After detailed analysis it is concluded that HAdFA performs better than HFPA. It is also demonstrated that adaptive strategies employed in FA helped HAdFA to achieve best optimal solutions. Future scope of this investigation is detailed in Chapter 7. The next Chapter, reports the study and findings of an FMS.

CHAPTER 5
COMBINED OBJECTIVE OPTIMIZATION OF FLEXIBLE MANUFACTURING SYSTEM SCHEDULING

5.1 INTRODUCTION

Flexible Manufacturing Systems (FMS) have been around since the 1960s and have become increasingly important in the manufacturing industry, especially in the current era of Industry 4.0. An FMS is a highly automated and computer-controlled system that is designed to produce a variety of products with minimal setup time and changeovers. FMS has become an essential part of modern manufacturing because it allows for flexibility in the production process, which is crucial in meeting the demands of consumers in a rapidly changing marketplace. FMS can accommodate a wide range of product designs and specifications, allowing manufacturers to produce customized and personalized products quickly and efficiently.

Optimal scheduling is crucial to the success of Flexible Manufacturing Systems (FMS) as it directly impacts the system's efficiency, productivity, and profitability. FMS are designed to produce a variety of products with minimal setup time and changeovers, and optimal scheduling helps to achieve this objective by ensuring that the machines are used in the most efficient manner possible. This results in increased productivity, reduced lead time, increased throughput, improved quality, and reduced costs. By optimizing scheduling, FMS can remain competitive in an increasingly challenging manufacturing environment.

This chapter details about the optimal scheduling of FMS taken from Jerald et al [161] which comprises of two test instances for 43 jobs 16 machines and 80 jobs and 16 machines. The objectives of the research are to combinedly optimize both machine idle time and the total penalty cost. Three meta heuristics viz., GAPSOTS, HFPA, HAdFA are developed to achieve combined optimal scheduling of proposed FMS. The results and performance comparison are given in following subsequent sections.

5.2 DESCRIPTION OF FMS TEST INSTANCES

The following are the problem context, premise, and goal of the current work:

Figure 5.1 shows the set-up of FMS taken for this study. A total of five Flexible Machine Cells (FMC) is there and each FMC has dedicated CNC machines which is shown in Table 5.1. The FMCs also has a tool magazine which is self-sufficient, An ATC to change the tools automatically, and an automatic pallet changer. There will be specified number of robots to move the materials in between the machines during operations. There is also a loading and unloading station. The unfinished products are stored in automatic storage and retrieval system (AS/RS). There are two Automated Guided Vehicles (AGVs) which connects all FMCs and helps in moving finished/ semi-finished products among the cells.

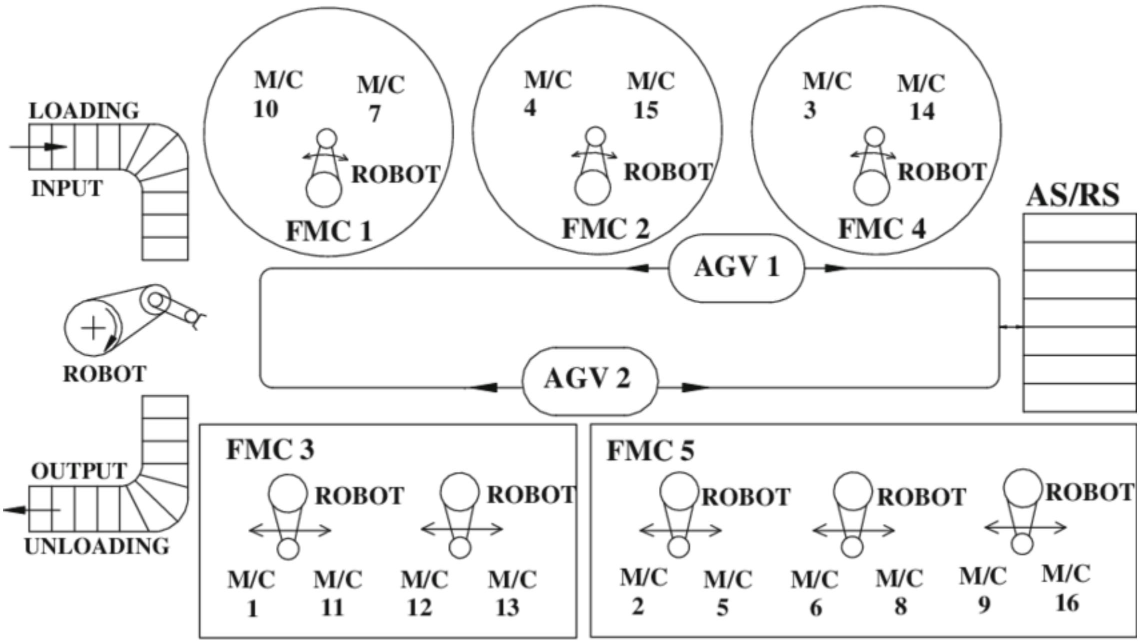


Figure 5.1 FMS configuration

Table 5.1 Details of FMC and its Dedicated Machines

FMC number	Machine number
1	M/C 7, M/C 10
2	M/C 4, M/C 15
3	M/C 1, M/C 11, M/C 12, M/C 13
4	M/C 3, M/C 14
5	M/C 2, M/C 5, M/C 6, M/C 8, M/C 9, M/C16

Following are the assumptions made for this work

1. The tool magazine's tool combinations can be utilized to create 40 to 80 different kinds of products.
2. Each product type or variety has its own processing order, batch size, deadline, and fine for missing the deadline.
3. A given machine has a processing time for each process in the processing process.

4. A randomly generated product mix (as given in Table 5.2) reflects the demand of the market at the time.

The FMS optimal scheduling is done for Combined Objective Function (COF) which minimizes the idle time of a machine and total penalty cost.

$$COF = \left[\omega^* \frac{Total\ Penalty\ Cost(TPC}{Maximum\ Permissible\ Penalty} + \omega^* \frac{Total\ Machine\ Idle\ Time\ (TMI)}{Total\ elapsed\ time}\right] \tag{5.1}$$

Where ω^* is weight factor and 0.5 is its value.

Table 5.2 Product mix for 43 jobs 16 machines

	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

5.3 INPUT DATA FOR FMS SCHEDULING

Three new novel meta- heuristic are proposed in this study to perform combined optimal scheduling of FMS. The working procedure of GAPSOTS, HFPA and HAdFA are already explained in detail Chapter 3- Soft Computing Techniques. Four benchmark test instances are taken from Jerald et al [161] and Nidhish [162] for studying the combined objective function. Test instance 1 has 10 jobs x 8 machines, Test instance two has 20 jobs x 15 machines, Test instance 3 has 43 jobs and 16 machines. Table 5.2 and 5.3 shows the product mix along with the processing machine time, due dates, penalty cost including the batch sizes.

Table 5.3 Product Mix for 80 jobs and 16 machines

Part no.	Processing sequence – M/c No. & processing time in min.	Deadline (days)	Batch size (Nos.)	Penalty cost (Rs./unit/day)
1	{6,1},{7,1},{8,1},{10,2}	17	150	1.00
2	{2,1},{6,1},{8,1},{9,2},{14,4},{16,2	17	200	1.00
3	{8,1},{11,3},{13,4}	14	800	1.00
4	{9,4}	26	700	2.00
5	{4,5},{5,3},{15,4}	11	150	1.00
6	{6,5},{14,1}	16	700	1.00
7	{3,5},{6,3},{16,5}	26	250	2.00
8	{5,4},{6,5},{8,1}	26	850	2.00
9	{4,1},{5,5},{8,1},{11,1}	1	100	0.00
10	{2,2},{9,1},{16,4}	20	150	2.00
11	{8,4},{12,2}	1	250	1.00
12	{6,2},{8,4},{10,1}	19	1000	3.00
13	{6,1},{7,5},{10,4}	25	700	4.00
14	{4,2},{5,3},{6,2},{15,2}	22	1000	4.00
16	{5,3}	27	750	3.00
15	{5,4},{8,3}	15	700	5.00
17	{3,1},{6,4},{14,1}	20	650	4.00
18	{9,2},{16,3}	24	250	5.00
19	{4,1},{5,5},{6,2},{8,2},{15.5}	5	450	1.00
20	{8,2},{11,4}	11	50	5.00
21	{4,5},{5,5},{6,2},{8,2},{15,5}	16	850	3.00
22	{12,5}	24	200	5.00
23	{4.2},{5,1},{6,5},{8,4}	14	50	4.00
24	{8,4},{11,4},{12,5},{13,4}	7	200	5.00
25	{7,3},{10,2}	24	350	1.00
26	{10,2}	27	450	0.00
27	{8,5},{11,5},{12,4}	22	400	1.00
28	{2,1},{8,1},{9,2}	3	950	5.00
29	{4,1},{5,5}	7	700	1.00
30	{11,3},{12,5}	18	1000	1.00
31	{8,2},{10,2}	2	800	2.00
32	{2,3},{6,4},{9,3}	15	800	1.00
33	{5,4},{6,5},{15,3}	27	500	4.00
34	{3,2},{6,2}	12	300	4.00
35	{3,4},{14,1}	9	900	2.00
36	{3,2}	20	700	2.00
37	{1,5},{2,2},{6,3},{8,3},{9,2},{16,4}	22	250	4.00
38	{2,4},{8,3},{9,2},{16,5}	8	50	1.00
39	{6,5},{10,5}	9	500	1.00
40	{2,2},{6,4},{9,4}	7	250	5.00
41	{5,1},{8,2},{15,1}	22	800	4.00
42	{2,5},{6,4},{9,3}{16,1}	19	400	2.00
43	{1,3},{5,2},{6,2},{8,2},{15,3}	15	550	3.00

44	(2,5),(6,4),(9,3)	12	350	1.00
45	(16,3),(8,2),(2,3),(9,5)	15	400	3.00
46	(1,3),(12,5),(13,4)	8	250	4.00
47	(13,2),(12,3)`	7	440	2.00
48	(8,2),(16,3),(5.2)	10	350	2.00
49	(1,3),(11,5)	9	300	1.00
50	(16,2),(9,2),(2,1),(6,3)	8	300	1.00
51	(7,3),(10,2)	20	250	2.00
52	(4,1),(1,2)	16	300	3.00
53	(14,3)	10	275	4.00
54	(10,6),(7,2)	13	375	2.00
55	(16,3),(9,4),(6,2),(5,3)	15	220	5.00
56	(13,2),(1,7),(11,3)	12	200	3.00
57	(5,3),(6,2),(9,3),(2,1)	5	150	1.00
58	(7,5)	7	550	1.00
59	(10,4),(7,8)	8	150	2.00
60	(2,1),(9,3),(16,1)	17	500	1.00
61	(1,6),(13,2),(12,3)	24	100	2.00
62	(11,2),(13,4)	16	1000	2.00
63	(5,3),(2,11)	18	240	3.00
64	(13,2),(11,3)	27	800	1.00
65	(14,3),(3,11)	19	440	2.00
66	(4,4),(1,3)	14	320	2.00
67	(13,2),(1,3),(12,4),(11,3)	22	600	4.00
68	(16,2),(9,2),(8,1),(6,1)	14	700	1.00
69	(8,1),(9,2),(6,3),(5,3),(2,2)	16	150	2.00
70	(7,5),(10,1)	15	230	1.00
71	(3,14)	7	450	2.00
72	(11,6),(12,10)	18	570	3.00
73	(4,1),(1,5)	9	250	4.00
74	(16,3),(9,2),(2,2)	13	200	3.00
75	(16,1)	3	230	1.00
76	(1,2),(5,3),(12,1)	6	310	2.00
77	(2,2),(5,1),(6,11)	12	330	3.00
78	(9,3),(6,2),(5,3)	14	280	2.00
79	(2,1),(9,3)	14	210	1.00
80	(8,3),(9,3)	10	50	3.00

5.4 RESULTS AND DISCUSSIONS

The proposed algorithms GAPSOTS, HFPA and HAdFA are implemented to minimize the before said objectives combinedly. The performance comparisons are shown in Table 5.4 of three proposed algorithms with existing Particle Swarm Optimization (PSO), Memetic Algorithm (MA), Simulated Annealing (SA), Genetic algorithm (GA) [161], [163]. Among the three algorithms HFPA and HAdFA tend to optimize better than GAPSOTS. However, the GAPSOTS also exhibit near optimal COF. There is meagre difference in achieving optimal values by the proposed techniques. Nevertheless, HAdFA and HFPA again proves to yield better solutions than other algorithms. Bar chart comparisons are shown in Figure 5.2.

Table 5.4 Performance Comparison of Algorithms

Problem Size	Objectives	HADFA	HFPA	GAPSOTS	PSO [161]	MA [161]	SA [161]	GA [161]
10J x 8M	COF	0.12421	0.10012	0.16383	0.10096	0.1436	0.20869	0.15897
	Sequence	4 8 10 5 2 3 9 6 7 1	6 7 2 9 1 3 10 5 8 4	8 2 5 10 1 3 9 4 6 7	2 4 3 6 1 8 9 10 7 5	4 9 8 1 3 10 2 5 6 7	10 1 8 2 9 3 5 4 6 7	1 9 8 10 4 7 3 6 5 2
20J x 15M	COF	0.42315	0.654318	0.75469	0.539488	0.82417	1.08455	0.97418
	Sequence	3 7 15 11 14 13 6 10 18 20 9 12 4 8 1 5 2 19 16 17	1 20 10 11 18 14 4 12 17 8 9 7 5 19 6 13 15 2 3 16	18 16 8 4 17 12 14 13 1 6 11 7 5 15 3 9 2 10 19 20	7 6 20 14 9 10 12 19 8 3 11 15 16 13 4 1 18 5 17 2	16 3 7 20 13 18 6 4 5 1 17 15 14 10 9 8 19 2 11 12	17 16 10 19 6 12 4 2 7 11 20 18 5 8 15 13 3 1 14	15 18 19 16 6 3 4 11 2 20 17 5 10 12 14 8 1 9 7 13
43J X 16M	COF	0.10589	0.1123	0.11967	0.2646	0.35136	0.45936	0.42898
	Sequence	12 27 2 21 4 28 3 31 15 16 13 6 24 19 41 42 40 36 33 7 43 9 37 23 11 39 35 38 22 17 32 14 26 1 20 29 10 18 5 8 25 34 30	20 30 4 23 16 34 2 39 8 11 37 31 43 27 15 7 19 9 1 24 40 38 3 32 25 35 5 22 41 26 29 17 42 28 12 18 33 13 6 21 36 14 10	30 29 8 26 13 31 38 24 37 20 33 7 11 36 23 12 39 18 34 42 35 21 28 36 17 25 5 4 15 22 9 40 2 1 27 14 10 32 19 41 16 43	40 22 41 39 1 30 10 27 24 33 4 37 19 43 23 25 11 5 17 32 15 31 9 42 29 34 28 12 20 2 36 38 8 6 7 16 26 14 13 21 18 35 3	3 1 29 22 30 42 35 28 9 20 7 41 2 5 34 21 13 4 6 39 36 25 37 17 27 15 40 38 43 33 31 26 32 14 24 10 12 16 18 19 11 8 23	34 19 26 29 25 3 35 21 37 2 15 8 23 40 27 6 22 33 4 11 39 18 42 24 32 1 9 16 36 43 12 38 13 5 28 10 41 17 31 20 30 7 14	4 3 6 17 43 19 8 31 23 13 16 37 39 12 15 41 32 18 30 22 11 27 42 7 26 38 14 1 40 36 9 10 5 28 29 35 2 24 33 34 20 25 21
80J x 16M	COF	0.0754632	0.082341	0.08537	-	0.1067266	0.14738715	0.0959237
	Sequence	19 44 76 69 22 56 27 30 23 71 59 60 45 7 16 49 42 25 24 55 17 79 61 6 36 51 8 34 68 35 62 70 15 57 12 63 53 20 72 26 46 52 5 77 66 21 50 33 1 10 58 65 80 28 54 73 13 48 9 41 4 78 64 37 2 67 14 32 43 74 31 11 47 40 75 3 29 38 18 39	7 48 20 41 37 54 74 21 70 57 55 46 26 34 69 62 49 76 53 59 31 1 58 43 40 42 44 5 61 23 16 39 63 30 56 28 38 14 2 8 73 13 19 75 68 3 77 65 51 32 25 22 18 15 10 80 9 6 71	19 63 31 24 38 51 5 69 49 37 22 45 59 46 23 74 68 7 66 13 53 78 40 67 44 4 62 58 11 65 18 75 79 60 17 77 61 71 57 8 35 15 26 50 32 27 64 39 30 10 6 55 25 1 16 28 29 54 76 80 14 73 21 33 47 20 2 70 56 52 42		40 49 23 9 30 11 18 68 48 15 67 6 5 19 28 21 3 46 10 51 75 69 79 36 54 31 72 44 25 77 65 38 27 59 32 56 55 58 78 42 12 76 74 7 16 61 13 8 66 45 41 35 53 52 26 70 64 43 14 2 57 33	24 1 37 77 3 53 20 59 39 19 28 51 16 32 66 47 52 36 49 14 55 65 76 10 57 17 61 72 74 4 43 50 78 48 58 31 67 80 45 15 34 2 12 63 75 23 70 29 35 18 40 68 73 9 11 71 27 62 56 5	34 57 62 76 6 22 67 10 8 72 68 61 13 20 53 15 26 29 9 25 31 58 18 64 41 3 7 33 49 60 4 50 54 24 47 71 55 70 45 40 37 65 51 73 12 16 56 5 28 39 19 75 30 80

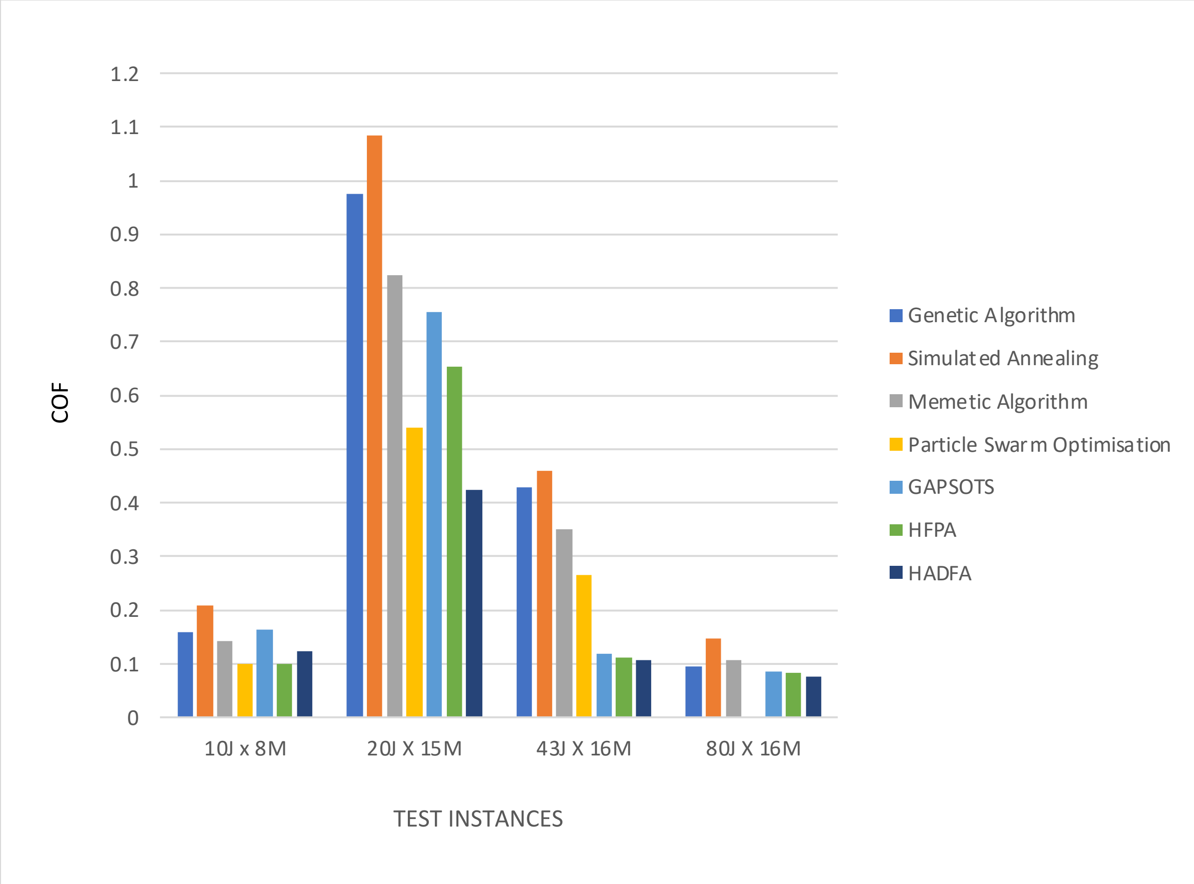


Figure 5.2 Bar chart comparisons of Algorithms

5.5 SUMMARY

This chapter detailed the Combined Objective Optimization (COF) for an FMS setup generated by Jerald et al. Four test instances with variable jobs ranging from 10 to 80 and machines ranging from 8 to 16 are studied and the performances are analyzed. The COF is to minimize machine idle time and total penalty cost combinedly. The test instances are tested by proposed heuristics GAPSOTS, HFPA and HAdFA. For all four test instances HAdFA yielded the best optimal value closely followed by HFPA and GAPSOTS. This research demonstrates that the methods created here may be effectively customized to work with any FMS that has numerous components and machines that are subject to multiple objective functions. Future components will include load/unloading stations' accessibility and handling durations, robots, and AGVs.

CHAPTER 6

CONCURRENT SCHEDULING OF MACHINES AND AUTOMATIC GUIDED VEHICLES

6.1 INTRODUCTION

This chapter details about the concurrent scheduling of machines and AGVs for different FMS layouts. FMS scheduling problem considering the elements such as tools and AGVs are very complex and hence powerful evolutionary algorithms are essential for solving these problems.

Simultaneous scheduling of machines and AGVs is a complex optimization problem that involves coordinating the utilization of resources within a manufacturing or warehousing environment. The goal is to find the optimal schedule for both machines and AGVs in order to maximize production efficiency, reduce idle time, and minimize overall costs. This requires taking into account a variety of factors such as machine and AGV capacities, production requirements, available raw materials, and the need to ensure that all tasks are completed within specified time windows. To achieve this, sophisticated algorithms are used to generate schedules that meet these requirements while ensuring that the system is working at maximum efficiency. The scheduling of machines and AGVs concurretly is a critical aspect of modern manufacturing and warehousing operations and is essential for achieving high levels of productivity, quality, and customer satisfaction

Both, machine scheduling and AGV scheduling are interrelated problems. An FMS will be able to perform all of the tasks assigned in shortest time span when AGVs and machines are scheduled together. Scheduling of AGVs play a keyrole in performance improvement of a FMS.

Scheduling optimization is done to optimise certain objectives which are decided upon the customer needs and requirements, market circumstances, customized needs of the customer. In order to solve the concurrent scheduling of AGVs and machines , the researcher proposes three new novel meta-heuristics named GAPSOTS, HADFA and HFPA for single objective of minimizing makespan and for multi objectives to minimize makespan, mean tardiness and mean flow time. The adequacy and effectiveness of the proposed approaches can be assessed by comparing the outcomes with those of existing approaches found in the literature.

Following are the assumptions considered in this study

- A machine cannot be operated for more than one operation at once.
- Each task consists of a number of operations, each with a specific order of priority.
- Transport timings for AGVs and processing times for operations are known.
- Recharging time of AGVs are negligible.
- Deadlock in operating AGVs is not taken into account.
- It is presumed that all machines, tools, jobs and AGVs are available initially.
- Loading and unloading times of AGVs are not considered.
- Pre-emption of jobs, tools and vehicles are not allowed
- Tool set up time and transfer times are not considered.

6.2 METHODOLOGICAL STEPS TO SCHEDULE MACHINES AND AGVS SIMULTANEOUSLY

Step1: The job sets, travel time matrix of AGVs, solution vectors are to be entered initially.

Step2: Proposed algorithm parameters are read following each other.

Step2a: Assign positions to jobs, operations, machines and AGVs.

Step3: Implement proposed meta-heuristic.

Step 3.1: population initialization

Step 3.2: Evaluate each vector for objective criteria.

Step4: When an operation of a job is completed, any available AGV is called upon to transfer the job to next location as per job's schedule.

Step5: The job will wait in the buffer if the assigned machine is busy, and it gets loaded when the machine becomes free.

Step6: Start the operation on the machine.

Step7: Check all operation are completed on the particular job, if not start from step 4.

Step8: Compute the objective function values upon completion of all operations.

6.3 MINIMIZING MAKESPAN – SINGLE OBJECTIVE APPROACH

In this section, Machines and AGVs are scheduled concurrently for different FMS layouts with an objective to minimize makespan. The input data , layouts of FMS are detailed subsequently in this section.

1. The first objective function minimizing makespan is calculated by following equations

$$O_{jk} = T_{jk} + P_{jk} \tag{4.1}$$

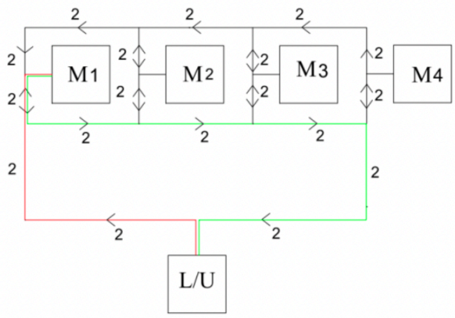
where O_{jk} is completion time required for an operation; T_{jk} is travelling time for AGVs; P_{jk} is processing time of an operation; k^{th} operation for j^{th} job.

∴ Time taken for a job to complete is $C_j = \sum_{j=1}^{n} O_{jk}$ (4.2)

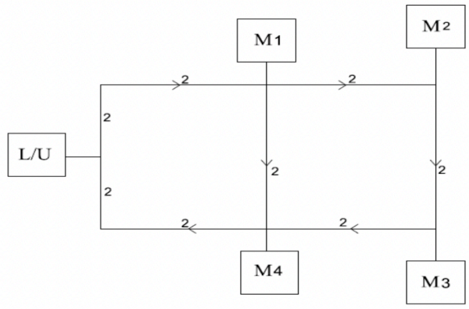
$$\text{Makespan} = \text{Max} (C_1 , C_2, C_3 \ldots C_n) \tag{4.3}$$

6.3.1 FMS Setup Considered in this Study

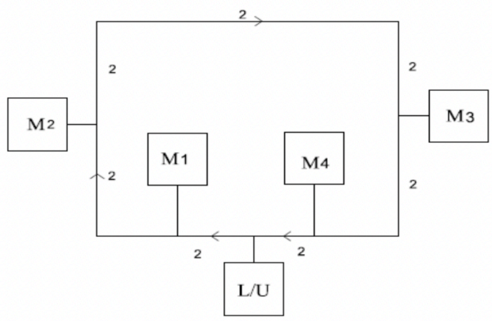
The FMS considered in this work has four machines having computer numerical machines (CNCs), each CNC has separate tool magazine which is self-sufficient, one automatic tool changer (ATC) and one automatic pallet changer (APC). Four different layouts are considered for optimal scheduling of machines and AGVs and it is shown in Figure 6.1.the travelling distances are shown in the Figure 6.1. AGV move with a speed of 40 m/min. Travel time matrix is given in Table 6.1 including loading and unloading. In Figure 6.1, L/U refers Load Unload Stations ; and M1,M2......refers Machines in order. The job set data can be found in Bilge and Ulusoy [164]



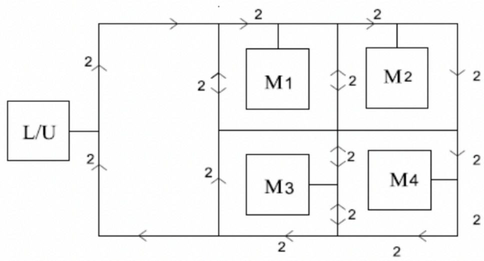
(a) Layout 1



(b) Layout 2



(c) Layout 3



(d) Layout 4

Figure 6.1 Different Layouts of FMS

Total of 40 problems are developed with the help of job set data and travel time matrix (TTM). For further calculation a second set of data is developed by having processing time doubled or tripled. In both the cases TTM is halved. Therefore in total 82 problems are considered in this study to minimize makespan. Further details can be acquired from Bilge and Ulusoy [164]. In this study, two conditions are considered and defined as travelling time/ processing time (t/p ratio).

Case i) First data set formed with actual processing and travel time and t/p > 0.25 (40 problems in total)

Case ii) Second data set formed with doubled or tripled processing time and half travel time and t/p < 0.25. (42 problems in total)

Table 6.1 Travel Time Matrix

Layout 1	From-To	L/U	M1	M2	M3	M4
	L/U	0	6	8	10	12
	M1	12	0	6	8	10
	M2	10	6	0	6	8
	M3	8	8	6	0	6
	M4	6	10	8	6	0
Layout 2	From-To	L/U	M1	M2	M3	M4
	L/U	0	4	6	8	6
	M1	6	0	2	4	2
	M2	8	12	0	2	4
	M3	6	10	12	10	2
	M4	4	8	10	12	0
Layout 3	From-To	L/U	M1	M2	M3	M4
	L/U	0	2	4	10	12
	M1	12	0	2	8	10
	M2	10	12	0	6	8
	M3	4	6	8	0	2
	M4	2	4	6	12	0
Layout 4	From-To	L/U	M1	M2	M3	M4
	L/U	0	4	8	10	14
	M1	18	0	4	6	10
	M2	20	14	0	8	6
	M3	12	8	6	0	6
	M4	14	14	12	6	0

6.4 MULTI-OBJECTIVE APPROACH

In addition, to the objective minimizing makespan, two more objectives are solved for scheduling machines and AGVs simultaneously .

1. The second objective function, mean flowtime (F_j) is the difference between the completion time and the arrival time of the job 'j' and is given by Equation 4.5.

   $F_j = C_j - A_j$ (4.4)

   where C_j is completion time of job 'j' and A_j is time taken for arrival of job 'j'

   $\therefore$ Mean Flow time $= \frac{1}{n}\sum_{j=1}^{n} F_j$ (4.5)

2. The third objective function mean tardiness is given by Equation 4.6. Tardiness is the difference between completion time and due date.

   Mean Tardiness $= \frac{1}{n}\sum_{j=1}^{n} T_j$ (4.6)

   where T_j is tardiness

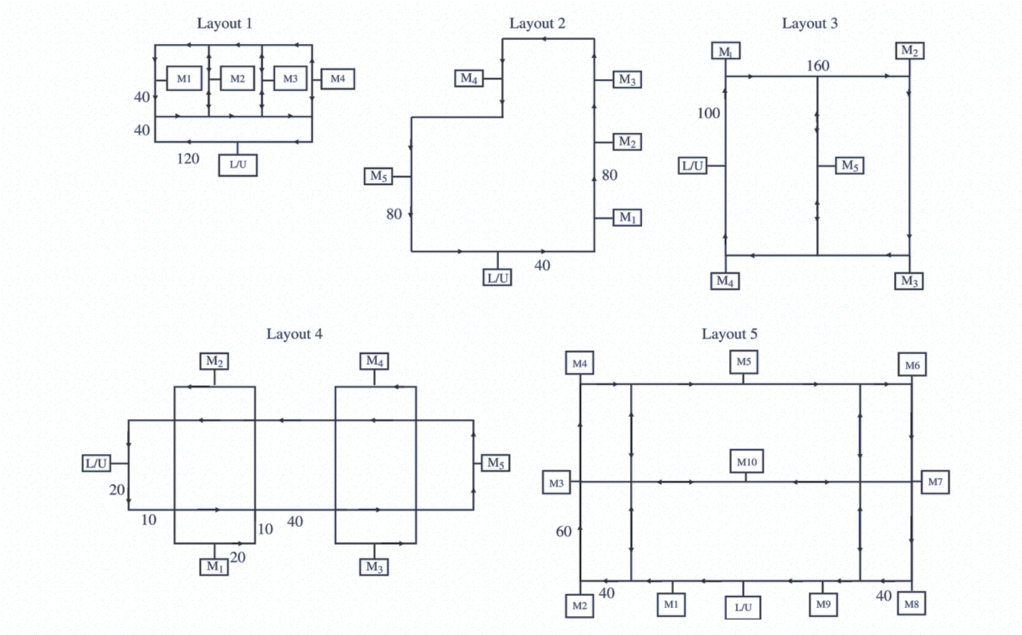


Figure 6.2 Different FMS layout

6.4.1 Input Data of FMS setup considered

The researcher tested the standard job shop benchmark instances for different FMS layouts. The benchmark data is available in https://github.com/tamy0612/JSPLIB/tree/master/instances The testing was done on selected test instances and the layouts used are shown in Figure 6.2.and the TTM are given in Table 6.2. Due dates are assumed for the test instances along with two AGVs and it is expected that each work would take an average amount of time to complete.

Table 6.2 Travel Time matrices (TTM)

TTM1

	L/U	M1	M2	M3	M4
L/U	0	6	8	10	12
M1	12	0	6	8	10
M2	10	6	0	6	8
M3	8	8	6	0	6
M4	6	10	8	6	0

TTM 2

	L/U	M_1	M_2	M_3	M_4	M_5
L/U	0	3	5	7	9	14
M_1	15	0	3	5	8	12
M_2	13	15	0	3	6	10
M_3	11	13	15	0	4	8
M_4	7	9	11	13	0	5
M_5	4	6	8	10	13	0

TTM 3

	L/U	M1	M2	M3	M4	M5
L/U	0	3.5	7.5	12.5	12.5	8
M1	12.5	0	5	10	10	5.5
M2	12.5	15	0	6	10	10.5
M3	7.5	10	10	0	5	5.5
M4	3.5	6	10	15	0	10.5
M5	8	11.5	5.5	10.5	5.5	0

TTM 4

	L/U	M_1	M_2	M_3	M_4	M_5
L/U	0	2.5	4.5	4.5	6.5	5.5
M_1	4.5	0	3.5	3.5	5.5	4.5
M_2	2.5	3.5	0	5.5	7.5	6.5
M_3	6.5	7.5	5.5	0	3.5	2.5
M_4	4.5	5.5	3.5	3.5	0	4.5
M_5	5.5	6.5	4.5	4.5	2.5	0

TTM 5

	L/U	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
L/U	0	2	4	5.5	7	8	11	9.5	11	11	6.5
M1	11	0	3	4.5	6	7	10	8.5	10	10	5.5
M2	11	12	0	2.5	4	7	10	8.5	10	10	5.5
M3	9.5	10.5	4.5	0	2.5	5.5	8.5	7	8.5	8.5	4
M4	11	12	6	4.5	0	4	7	8.5	10	10	9.5
M5	8	9	11	9.5	11	0	4	5.5	7	7	6.5
M6	7	8	10	8.5	10	11	0	2.5	4	6	5.5
M7	5.5	6.5	8.5	7	8.5	9.5	4.5	0	2.5	4.5	4
M8	8	5	7	8.5	10	11	6	4.5	0	3	5.5
M9	2	3	5	6.5	8	9	12	10.5	12	0	7.5
M10	6.5	7.5	5.5	4	5.5	6.5	5.5	4	5.5	5.5	0

6.5 RESULTS AND DISCUSSIONS FOR SINGLE OBJECTIVE

The algorithms proposed in this study (GAPSOTS, HAdFA and HFPA) are tested on 82 data sets taken from Bilge and Ulusoy. Single objective to minimize makespan when both machines and AGVs are scheduled simultaneously is considered for this data set. Along with findings from previously used approaches, the outcomes of the proposed methods are reported in the following Sub Sections.

6.5.1 Results of Case (i) when t/p > 0.25

This section details the results of proposed GAPSOTS, HAdFA and HFPA compared with techniques found in literature. Table 6.3, 6.4, 6.5, 6.6 depicts the comparison with exiting techniques for Layout 1, Layout 2, Layout 3, Layout 4. The minimum makespan obtained is highlighted in bold. The bar charts from Figure 6.3-6.7 depicts the same results for the four layouts.

Table 6.3 Results Comparison for Layout 1 when t/p > 0.25

JOB NO:	STW [164]	UGA [165]	AGA [166]	PGA [167]	SALS [168]	FPA [169]	GAPSOTS	HFPA	HADFA
1	96	96	96	96	96	96	96	96	**96**
2	105	104	102	100	102	100	105	**100**	102
3	105	105	99	99	99	99	99	99	**85**
4	118	116	112	112	112	112	116	112	**85**
5	89	87	87	87	87	87	87	85	**80**
6	120	121	118	118	118	118	119	115	**114**
7	119	118	115	111	111	111	118	110	**98**
8	161	152	161	161	161	151	152	150	**146**
9	120	117	118	116	116	116	118	118	**115**
10	153	150	147	147	147	150	150	150	**146**

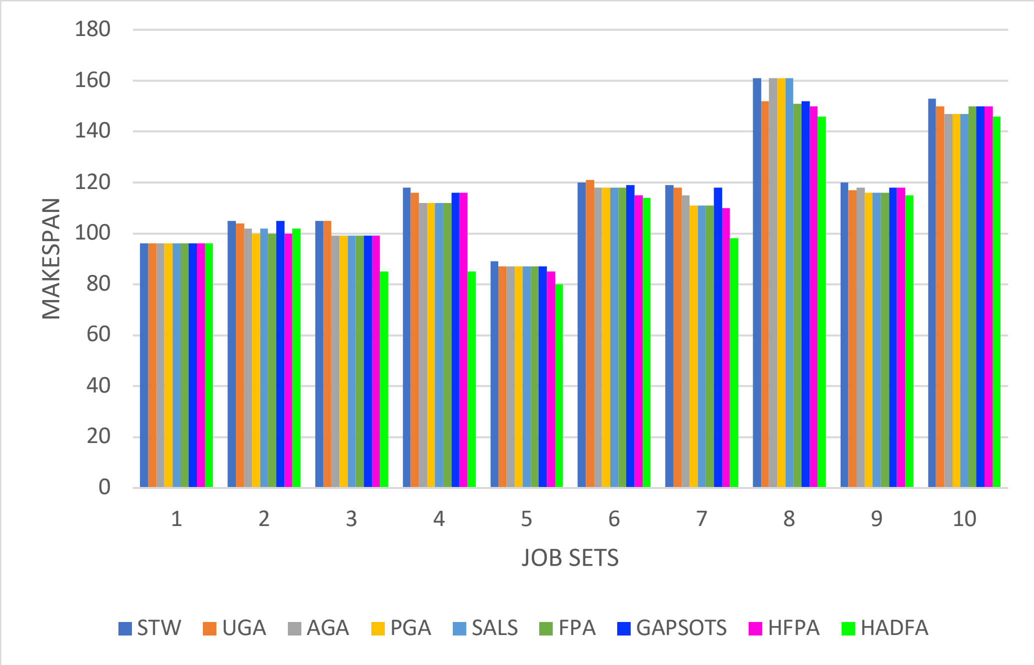


Figure 6.3 Results Comparison for Layout 1 when t/p > 0.25

From Table 6.3 it is inferred that HAdFA gives the best performance when compared to other proposed techniques and existing techniques from literature. HAdFA outperforms other techniques for 9 problems. For job set 2, HFPA has given minimum makespan. HFPA performed better than GAPSOTS for all job sets. GAPSOTS either gave similar results or worse results to existing values.

Table 6.4 Results Comparison for Layout 2 when t/p > 0.25

JOB NO:	STW	UGA	AGA	PGA	SALS	FPA	GAPSOTS	HFPA	HADFA
1	82	82	82	82	82	82	82	82	82
2	80	76	76	76	76	76	85	76	**70**
3	88	85	85	85	85	85	85	85	85
4	93	88	88	87	87	87	88	87	87
5	69	69	69	69	69	69	76	69	69
6	100	98	98	98	98	98	98	98	**93**
7	90	85	79	79	79	79	79	79	79
8	151	142	151	151	151	141	151	142	**141**
9	104	102	104	102	102	102	102	102	**99**
10	139	137	136	135	135	135	137	135	**131**

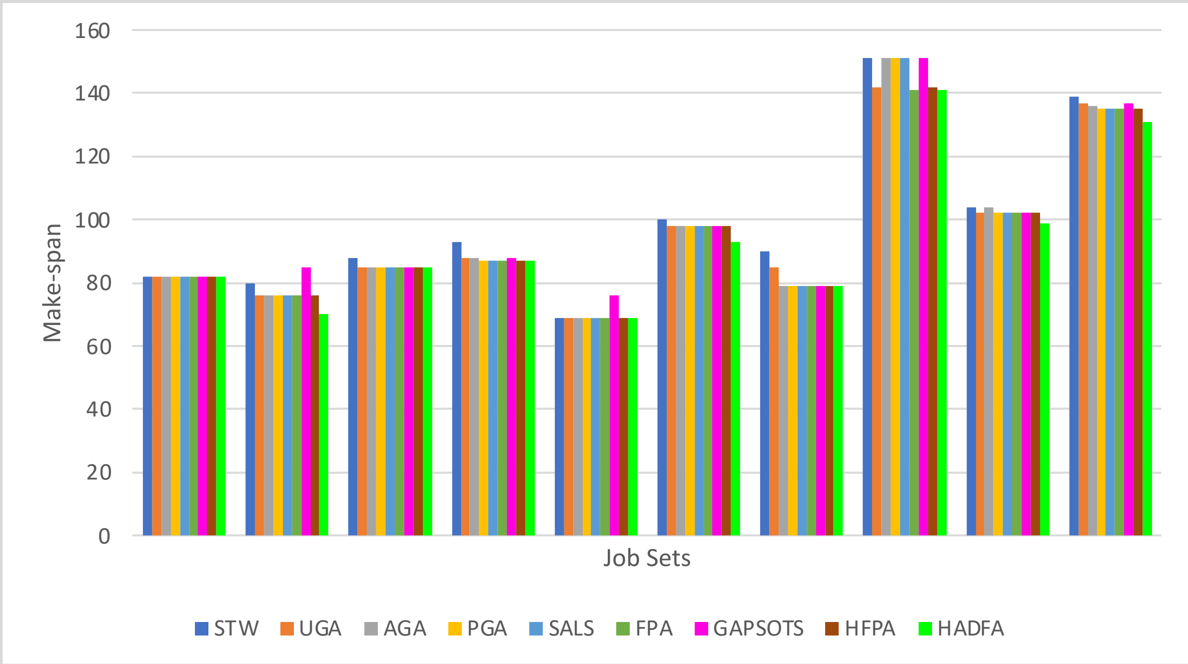


Figure 6.4 Results Comparison for Layout 2 when t/p > 0.25

From Table 6.4, it is deduced that all proposed algorithms are unable to give best results for all job sets. GAPSOTS gave slightly worse solution for Job set 2 and Job set 5. HFPA is unable to produce any better values than existing values. HAdFA is able to give new makespan values for Job set 2, 6, 8,9 and 10. HAdFA is consistently performing better than algorithms.

Table 6.5 Results Comparison for Layout 3 when t/p > 0.25

JOB NO:	STW	UGA	AGA	PGA	SALS	FPA	GAPSOTS	HFPA	HADFA
1	84	84	84	84	84	84	84	84	**82**
2	86	86	86	86	86	86	*98*	86	**85**
3	86	86	86	86	86	86	86	86	**78**
4	95	91	89	89	89	89	89	89	**69**
5	76	75	74	74	74	74	74	74	**72**
6	104	104	104	103	103	103	103	103	**101**
7	91	88	86	83	83	83	*94*	82	**80**
8	153	143	153	153	153	143	143	137	**137**
9	110	105	106	105	105	105	105	105	105
10	143	143	141	139	138	139	*149*	139	**119**

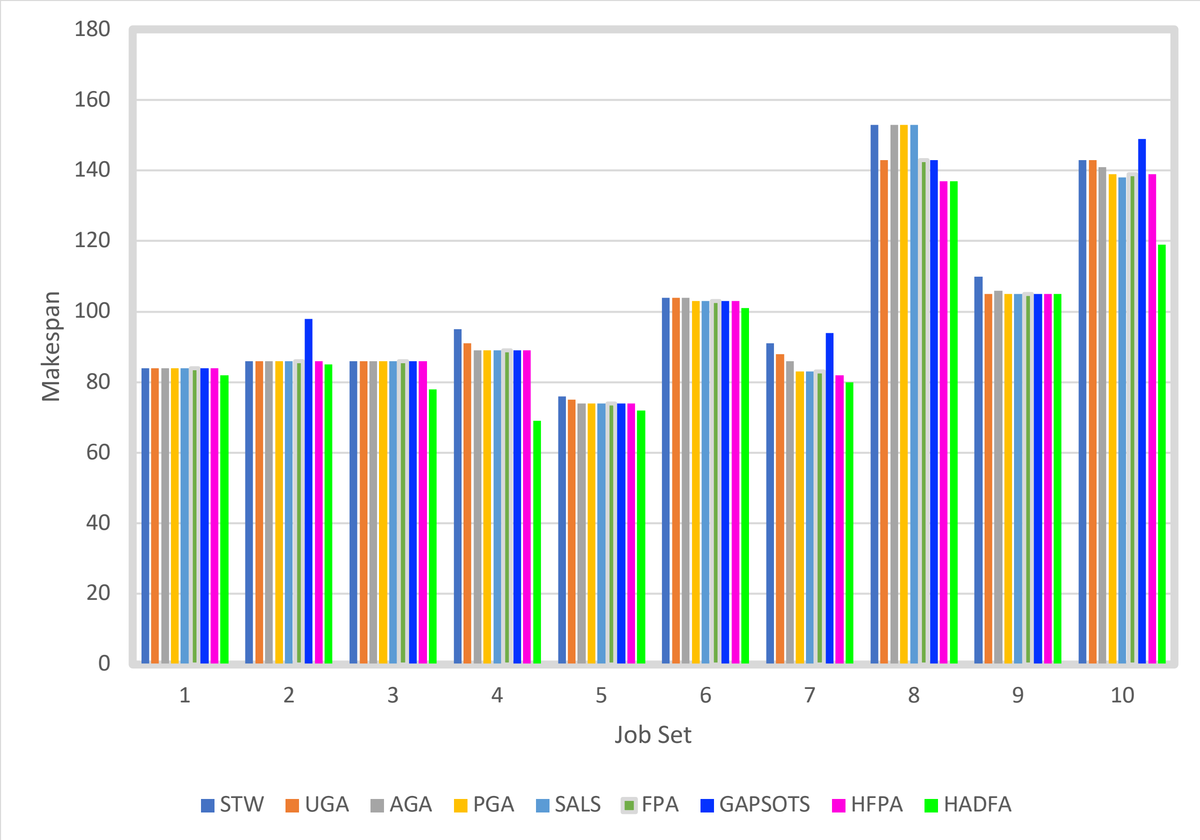


Figure 6.5 Results Comparison for Layout 3 when t/p > 0.25

It can be observed from Table 6.5, that for Layout 3 the proposed algorithms GAPSOTS,HFPA, HAdFA gives better results than existing algorithms for most of the job sets. Though GAPSOTS provides slightly worse results for Job set 2, 7 and 10 this only proves that evolution based algorithms are unable to procure best optimal solutions. HFPA gives best makespan than other existing methods for Job set 7, 8 and 10. HAdFA is the one which out performs other methodologies for all job sets except no.9 where the makespan is similar to PGA and SALS.

Table 6.6 and Figure 6.6. shows the performance comparison of Layout 4. It can be clearly observed HAdFA again outperforms other optimization techniques. New makespan values are found for all job sets. HFPA performed better than GAPSOTS for job set 4,6,7,8,9 and 10. GAPSOTS gives similar results of existing methodologies except Job set 6 where it gives slightly worse results.

Table 6.6 Results Comparison for Layout 4 when t/p > 0.25

JOB NO:	STW	UGA	AGA	PGA	SALS	FPA	GAPSOTS	HFPA	HADFA
1	108	103	103	103	103	103	103	103	**102**
2	116	113	108	108	108	108	108	108	**106**
3	116	113	111	111	111	111	111	111	**84**
4	126	126	126	126	121	126	126	121	**84**
5	99	97	96	96	96	96	96	96	**86**
6	120	123	120	120	120	120	124	120	**101**
7	136	128	127	126	126	126	127	126	**112**
8	163	163	163	163	163	153	163	151	**150**
9	125	123	122	122	120	122	123	120	**109**
10	171	164	159	158	159	158	165	157	**119**

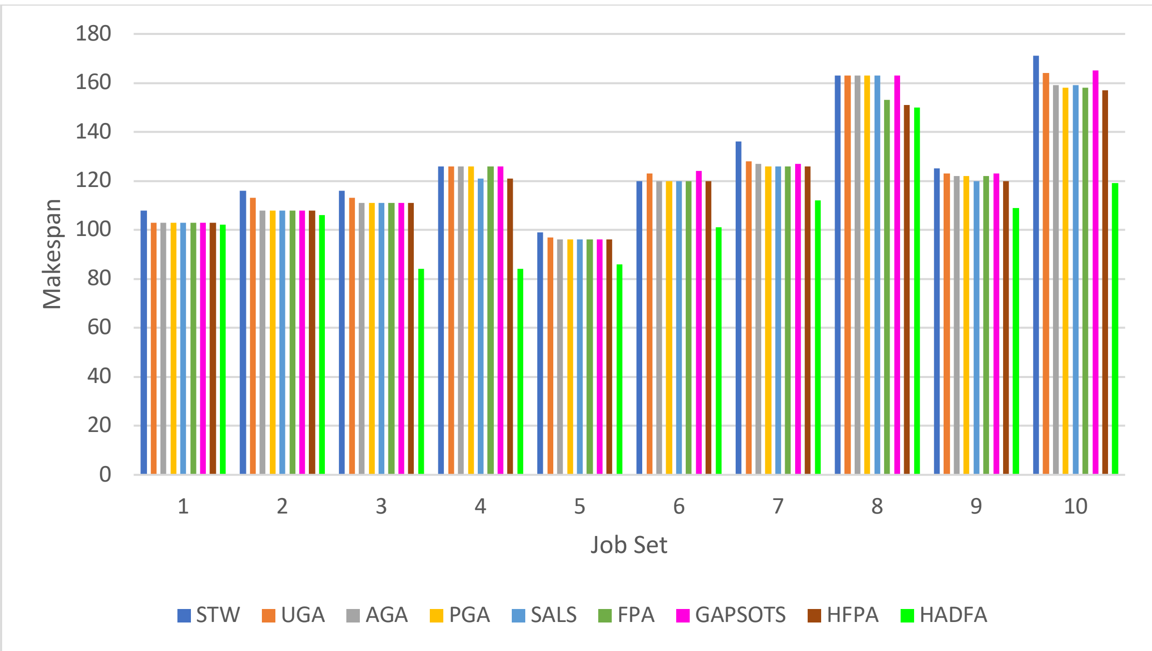


Figure 6.6 Results Comparison for Layout 4 when t/p > 0.25

6.5.2 Results of Case (ii) for t/p < 0.25

Table 6.7,6.8,6.9,6.10 and Figure 6.7,6.8,6.9,6.10 represents the Case (ii) where the travelling time to processing ratio is lesser than 0.25. (t/p < 0.25) for all four layouts.

Table 6.7 Results Comparison for Layout 1 when t/p < 0.25

JOB NO:	STW	UGA	AGA	PGA	TS SPMA	FPA	GAPSOTS	HFPA	HADFA
1	126	126	126	126	126	126	126	126	**126**
2	148	148	148	148	148	148	148	148	**148**
3	150	148	150	150	150	150	150	150	**143**
4	121	119	119	119	119	119	119	119	**116**
5	102	102	102	102	102	102	102	102	**102**
6	186	186	186	186	186	186	186	186	**173**
7	137	137	137	137	137	137	137	137	**137**
8	292	271	292	292	292	272	272	272	**271**
9	176	176	176	176	176	176	176	176	**161**
10	238	236	238	238	238	238	238	238	**224**

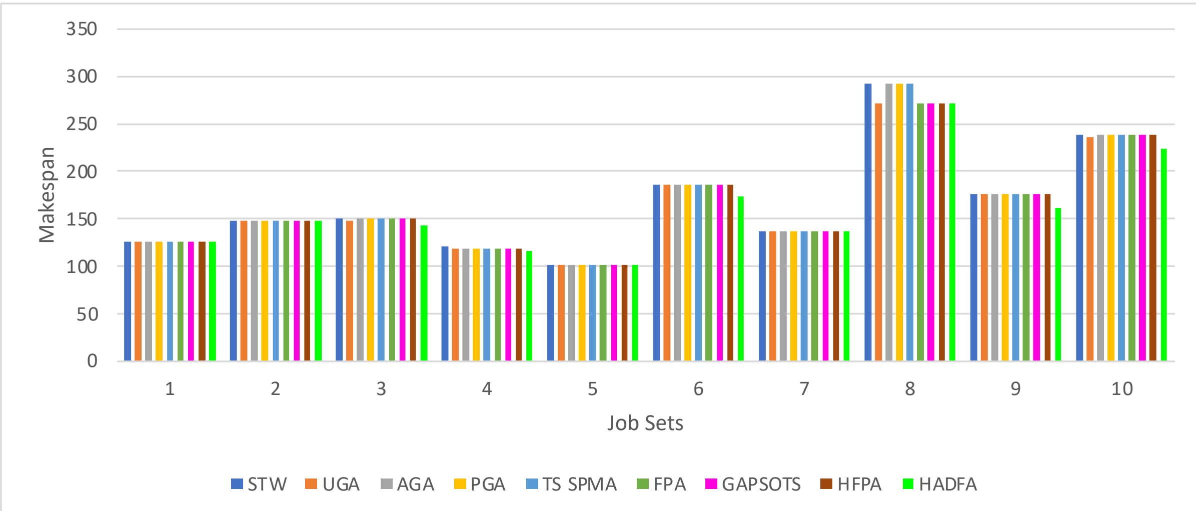


Figure 6.7 Results Comparison for Layout 1 when t/p < 0.25

Table 6.7 and Figure 6.7 illustrates the performance comparison of the proposed algorithms with existing algorithms. All three algorithms have achieved lower bound makespan values [164] for 4 problems out of 10 job sets. HFPA and GAPSOTS have executed similar makespan values. HAdFA has given new makespan values for job sets 3,4,6,8,9 and 10.

Table 6.8 Results Comparison for Layout 2 when t/p < 0.25

JOB NO:	STW	UGA	AGA	PGA	TS SPMA	FPA	GAPSOTS	HFPA	HADFA
1	123	123	123	123	123	123	**123**	**123**	**123**
2	143	143	143	143	143	143	**143**	**143**	**143**
3	148	145	145	145	145	145	**145**	**145**	**145**
4	116	114	114	114	114	114	**114**	**114**	**114**
5	100	100	100	100	100	100	**100**	**100**	**100**
6	181	181	181	181	181	181	**181**	**181**	**181**
7	136	136	136	136	136	136	**136**	**136**	**136**
8	287	268	287	287	287	267	**267**	**267**	**267**
9	174	173	173	173	173	173	173	173	**156**
10	236	238	236	236	236	236	236	236	**222**

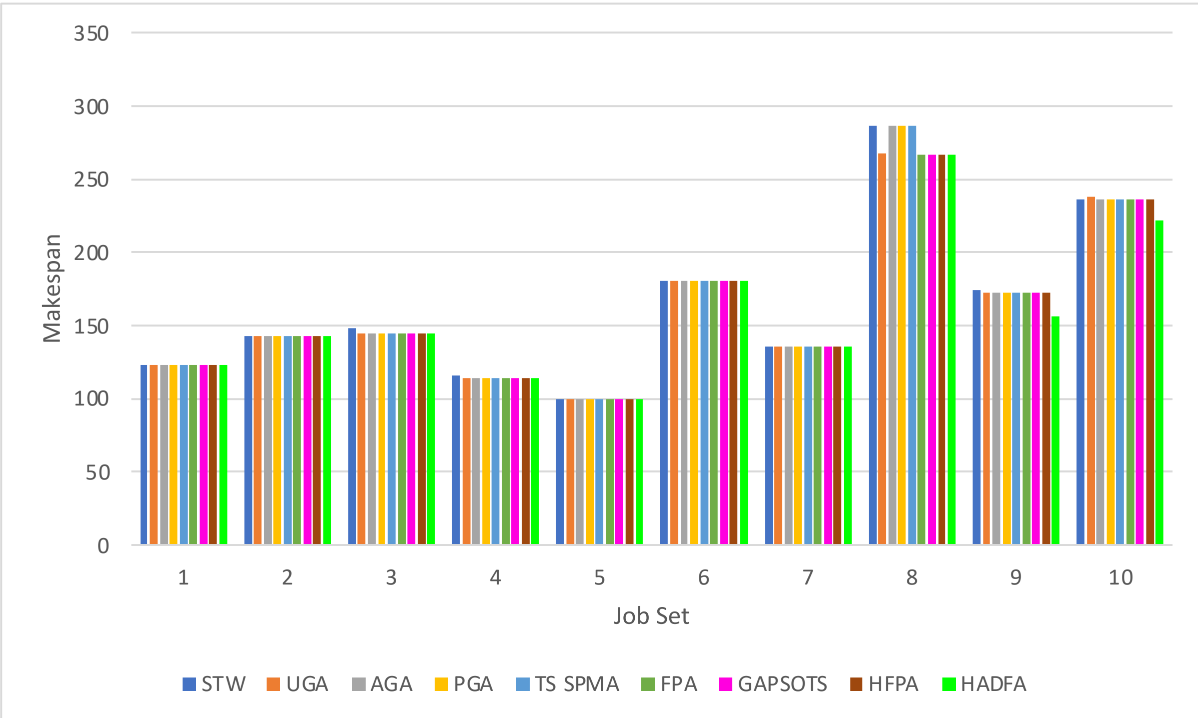


Figure 6.8 Results Comparison for Layout 2 when t/p < 0.25

The Table 6.8 illustrates that the proposed GAPSOTS, HFPA and HAdFA are able to achieve best makespan values for all the cases. HAdFA has performed much better than existing algorithms and gave new low bound makespan values of 156 and 222 for job nine and ten. Figure 6.8 depicts the same results in the form of bar chart.

Table 6.9 Results Comparison for Layout 3 when t/p < 0.25

JOB NO:	STW	UGA	AGA	PGA	TS SPMA	FPA	GAPSOTS	HFPA	HADFA
1	122	122	122	122	122	122	122	122	122
2	146	146	146	146	146	146	146	146	146
3	146	146	146	146	146	146	146	146	**141**
4	116	114	114	114	114	114	114	114	114
5	99	99	99	99	99	99	99	99	99
6	184	182	182	182	182	182	182	182	165
7	137	137	137	137	137	137	137	137	137
8	288	268	270	268	288	268	268	268	268
9	176	174	174	174	174	174	174	174	174
10	237	237	237	237	237	237	237	237	237

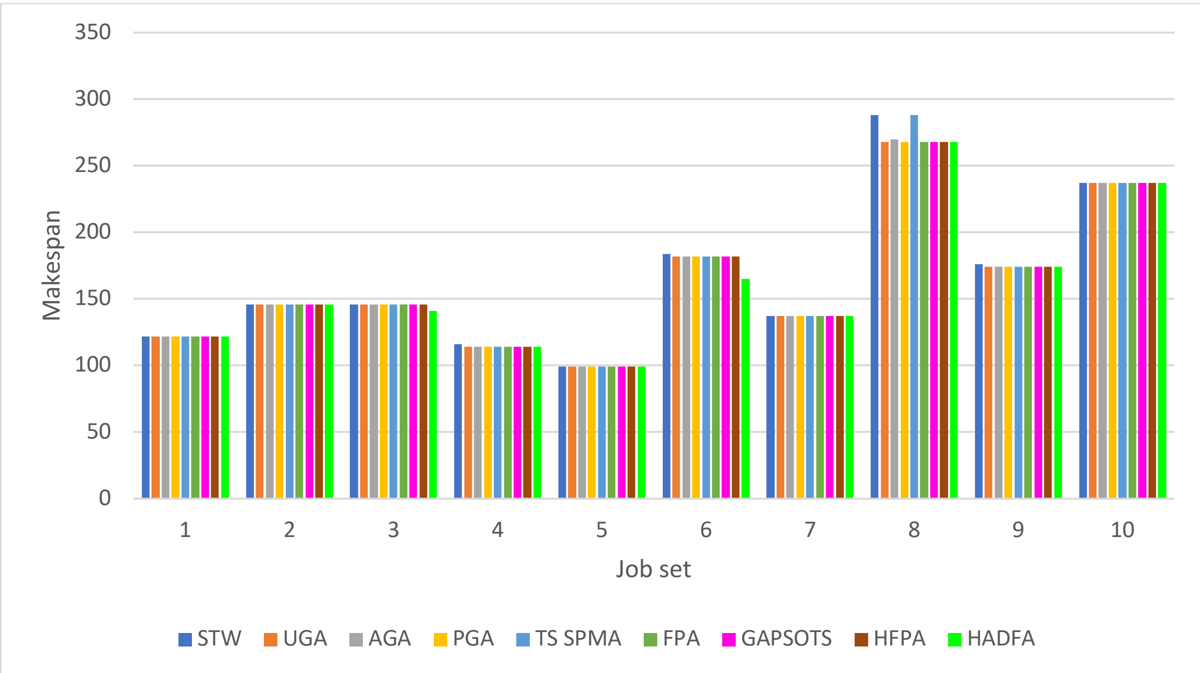


Figure 6.9 Results Comparison for Layout 3 when t/p < 0.25

From Table 6.9 it can be observed that all job sets have achieved same makespan values except job 3 of HAdFA, which has given new makespan value of 141 and it has been highlighted in bold numbers. GAPSOTS and HFPA gave similar results for all job sets.

Table 6.10 Results Comparison for Layout 4 when t/p < 0.25

JOB NO:	STW	UGA	AGA	PGA	TS SPMA	FPA	GAPSOTS	HFPA	HADFA
1D	124	124	124	124	124	124	124	124	124
3D	151	151	151	151	151	151	151	151	151
6D	185	184	184	184	184	184	184	184	184
7D	138	137	137	137	137	137	137	137	137
8D	293	273	293	293	293	273	*293*	273	**272**
9D	177	175	175	175	175	175	175	175	**159**
10D	240	244	240	240	240	240	243	240	240
2T	217	217	217	217	217	217	217	217	217
3T	221	221	221	221	221	221	221	221	221
4T	179	172	172	172	172	172	172	172	**171**
5T	154	148	148	148	148	148	148	148	148
7T	203	203	203	203	203	203	203	203	203

D represents the process time are doubled while travel times are halved
T represents the process time are tripled while travel times are halved

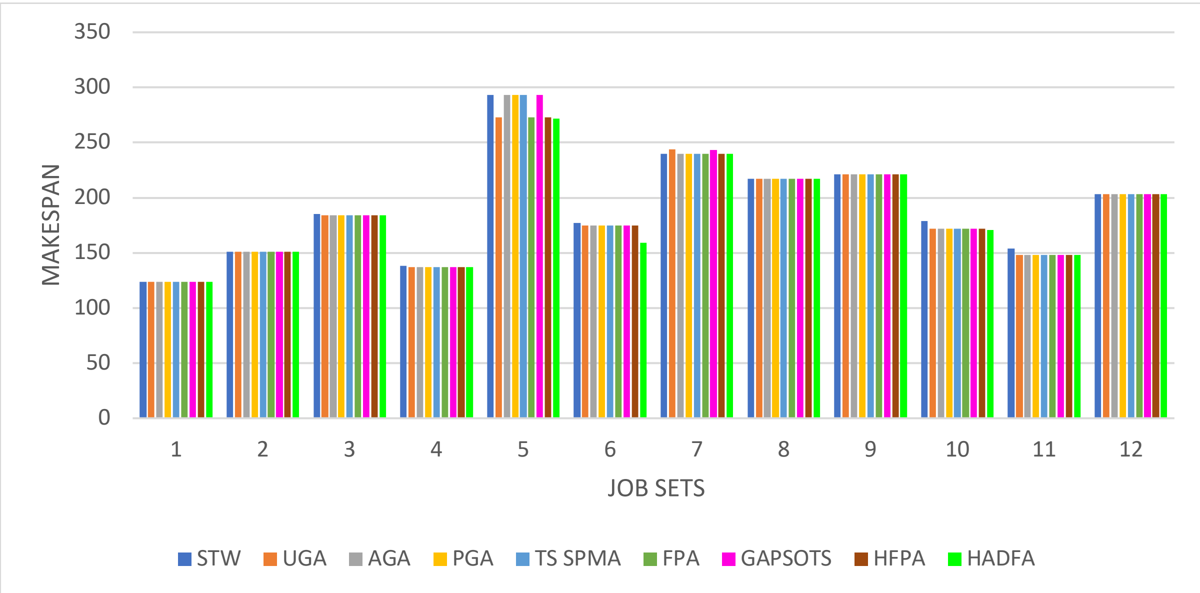


Figure 6.10 Results Comparison for Layout 4 when t/p < 0.25

The Layout 4 for ten job sets achieves same makespan for HAdFA like existing algorithms except 8D,9D and 4T. HAdFA is consistently proving that it is achieving new makespan values for at least 70% of the problems. This clearly indicates that the adaptive features integrated in classic Firefly Algorithm helps the problem to achieve new makespan results. GAPSOTS and HFPA gave similar results to that of existing techniques.

6.5.3 Comparison Analysis Among Proposed Algorithms

Case (i)

Table 6.11 compares the makespans of the three suggested techniques: GAPSOTS, HFPA and HAdFA for simultaneously scheduling machines, jobs, and AGVs for Case i for all layouts. Bold letters indicate the ideal values. Figure 6.11 displays the same data in a bar chart. From Table 6.11, it can be observed that HAdFA has given best makespan for 30 problems out of 40 for case (i). HFPA has given best makespan for one problem- Job set 2/ Layout 1 and also it has achieved similar makespan like HAdFA for 6 problems. GAPSOTS has not achieved any best makespan for any of the problems but has given similar makespan as that of HFPA for most of the problems. From this observation we can conclude that HAdFA gives the best makespan among the proposed algorithms.

Table 6.11 Makespan obtained by proposed algorithms for case (i)

Job Set	Layout 1			Layout 2			Layout 3			Layout 4		
	GAPSOTS	HFPA	HADFA	GAPSOTS	HFPA	HADFA	GAPSOTS	HFPA	HADFA	GAPSOTS	HFPA	HADFA
1	96	96	96	82	82	82	84	84	**82**	103	103	**102**
2	105	**100**	102	85	76	**70**	98	86	**85**	108	108	**106**
3	99	99	**85**	85	85	85	86	86	**78**	111	111	**84**
4	116	116	**85**	88	87	87	89	89	**69**	126	121	**84**
5	87	85	**80**	76	69	69	74	74	**72**	96	96	**86**
6	119	115	**114**	98	98	**93**	103	103	**101**	124	120	**101**
7	118	110	**98**	79	79	79	94	82	**80**	127	126	**112**
8	152	150	**146**	151	142	**141**	143	137	**137**	163	151	**150**
9	118	118	**115**	102	102	**99**	105	105	105	123	120	**109**
10	150	150	**146**	137	135	**131**	149	139	**119**	165	157	**119**

Figure 6.11 Makespan obtained by proposed algorithms for case i

Case (ii)

Table 6.12 compares the make-spans of the three suggested techniques: GAPSOTS, HFPA and HAdFA for simultaneously scheduling machines, jobs, and AGVs for Case ii for all layouts. Bold letters indicate the ideal values. Figure 6.12 displays the same data in a bar chart. From Table 6.12, it can be observed that HAdFA has given best makespan for 10 problems out of 40 for case (ii). HFPA and GAPSOTS have given similar results for all the problems. For case ii, HAdFA was unable to give best makespan for most of the problems.

Table 6.12 Makespan obtained by proposed algorithms for case (ii)

Job Set	Layout 1			Layout 2			Layout 3		
	GAPSOTS	HFPA	HADFA	GAPSOTS	HFPA	HADFA	GAPSOTS	HFPA	HADFA
1	126	126	126	123	123	123	122	122	122
2	148	148	148	143	143	143	146	146	146
3	150	150	**143**	145	145	145	146	146	**141**
4	119	119	**116**	114	114	114	114	114	114
5	102	102	102	100	100	100	99	99	99
6	186	186	**173**	181	181	181	182	182	**165**

Table 6.12 contd

7	137	137	137	136	136	136	137	137	137
8	272	272	**271**	267	267	267	268	268	268
9	176	176	**161**	173	173	156	174	174	174
10	238	238	**224**	236	236	**222**	237	237	237

Layout 4			
JOBSET	GAPSOTS	HFPA	HADFA
1D	124	124	124
3D	151	151	151
6D	184	184	184
7D	137	137	137
8D	*293*	273	**272**
9D	175	175	**159**
10D	243	240	240
2T	217	217	217
3T	221	221	221
4T	172	172	**171**
5T	148	148	148
7T	203	203	203

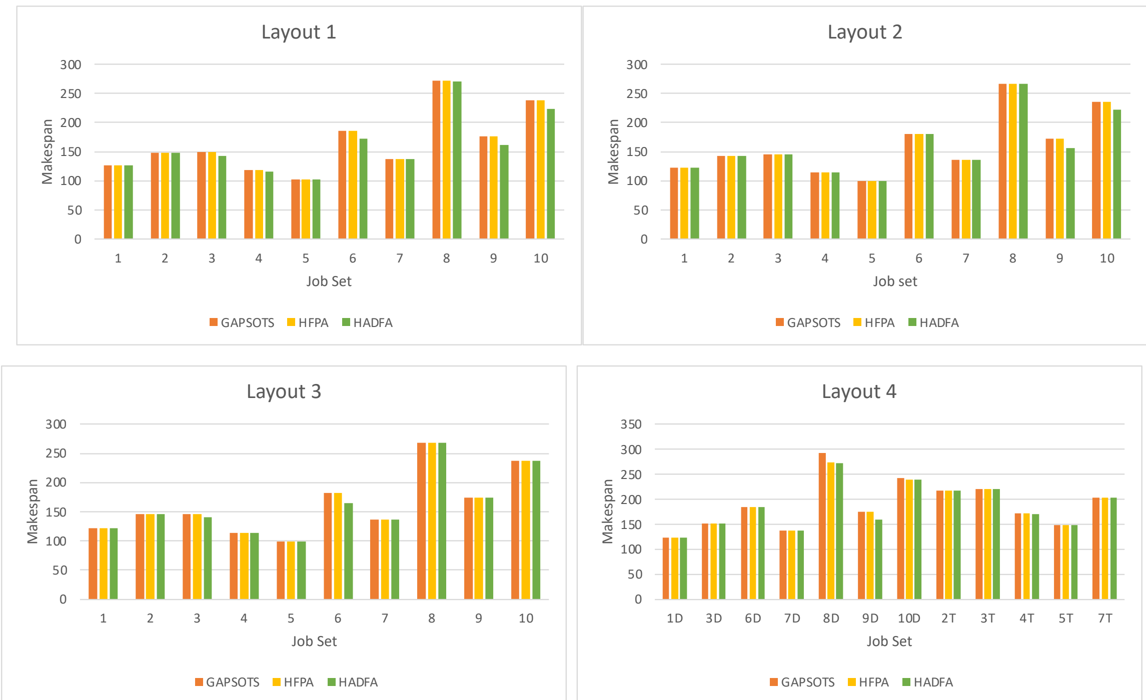


Figure 6.12 Makespan obtained by proposed algorithms for case ii

6.6 RESULTS AND DISCUSSIONS FOR MULTI OBJECTIVE

This section details about the performance comparison of HAdFA with HFPA for 5 test instances for simultaneous scheduling of machines and AGVs. Five different layouts (given in Figure 6.2) are chosen for this study. The aim is to minimize makespan , mean flowtime, mean tardiness. The researcher implemented HAdFA and HFPA for this study as its already proven that HFPA gives better results than proposed GAPSOTS. As there are no existing results for these job sets, comparison analysis with other techniques could not be done. The details of the test instances studied are given from Table 6.13 to 6.17.

Table 6.13 Test Instance 1

Layout used	1
Number of machines	4
Travel Time matrix	TTM 1
Job set details	Bilge and Ulusoy : job set 1/Layout 1
Number of Jobs	4
Total number of operations	13
Due date	20

Table 6.14 Test Instance 2

Layout used	2
Number of machines	5
Travel Time matrix	TTM 2
Job set details	JSP1 Bagchi
Number of Jobs	10
Total number of operations	50
Due date	37

Table 6.15 Test Instance 3

Layout used	3
Number of machines	5
Travel Time matrix	TTM 3
Job set details	G1 Lawrence 15x5 Instance
Number of Jobs	15
Total number of operations	75
Due date	65

Table 6.16 Test Instance 4

Layout used	3
Number of machines	5
Travel Time matrix	TTM 3
Job set details	G1 Lawrence 15x5 Instance
Number of Jobs	15
Total number of operations	75
Due date	65

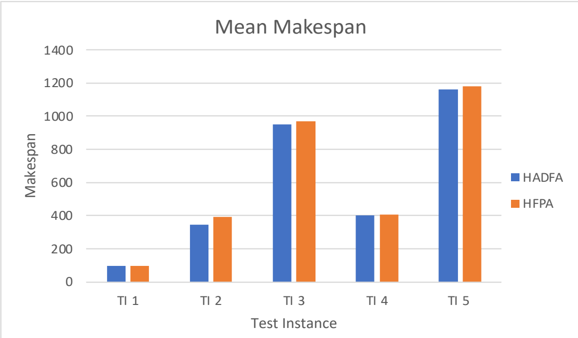
Table 6.17 Test Instance 5

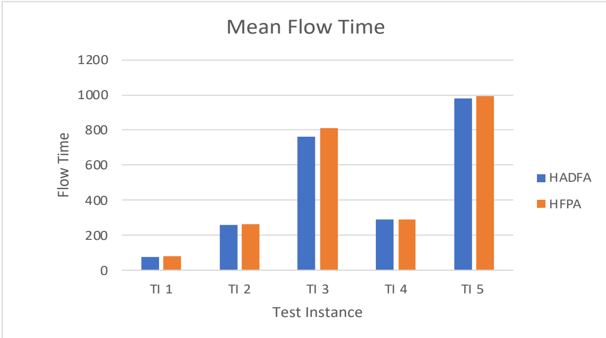
Layout used	5
Number of machines	10
Travel Time matrix	TTM 5
Job set details	A5 Lawrance 10x10 Instance
Number of Jobs	10
Total number of operations	100
Due date	115

Table 6.18 and Figure 6.13 depicts the results of performance metrics achieved by HAdFA and HFPA. The best results are given in bold.

Table 6.18 Performance outputs of HAdFA for multi-objectives

	Makespan		Mean Flow Time		Mean Tardiness	
	HADFA	HFPA	HADFA	HFPA	HADFA	HFPA
TI 1	**96**	96	**78.8**	82	**7**	8.5
TI 2	**344**	392	**257.1**	262	**101.4**	104
TI 3	**950.5**	971	**763**	809	277.34	**276**
TI 4	**403.2**	408	**290.02**	291	**98**	99
TI 5	**1162**	1180	**980**	995	**389**	390





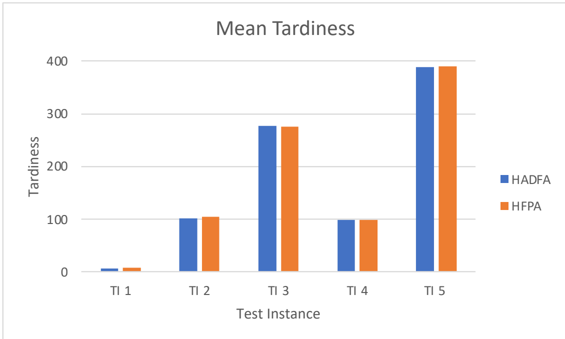


Figure 6.13 Performance outputs of HAdFA for multi-objectives

From Table 6.18 and Figure 6.13 it is observed that HAdFA gives superior results than HFPA which demonstrates that HAdFA is consistently achieving best performance metrics irrespective of problem sizes. The results also shows that multi-objective optimization can be accomplished by HAdFA effortlessly. HAdFA has given best results for all problems except TI 3 mean tardiness where HFPA has given best mean tardiness for that problem.

6.7 CONVERGENCE ANALYSIS FOR GAPSOTS, HAdFA AND HFPA

The convergence graphs of GAPSOTS, HFPA and HAdFA are displayed in Figure 6.14 for makespan of Job set 4, Layout 4. Table 6.19 demonstrates the time taken to converge, the iteration value when best makespan is obtained. The HAdFA takes more time to converge due to the adaptive features incorporated. HFPA converges earlier than GAPSOTS and HAdFA. GAPSOTS takes more time to converge but has taken a smaller number of iterations to get optimal value. This is because one iteration of GAPSOTS corresponds to 3 phases. This leads to the conclusion that evolutionary algorithms are complicated and time consuming process. The best makespan is achieved by HAdFA irrespective of extra time to converge and it is due to the adaptive features of HAdFA.

Table 6.19 Convergence Comparison of Proposed Algorithms

Algorithm	**Time (s)**	**Best Makespan**	**Iteration number when the best value is achieved**
GAPSOTS	2.216734	126	138
HFPA	0.156754	121	240
HADFA	0.245436	84	278

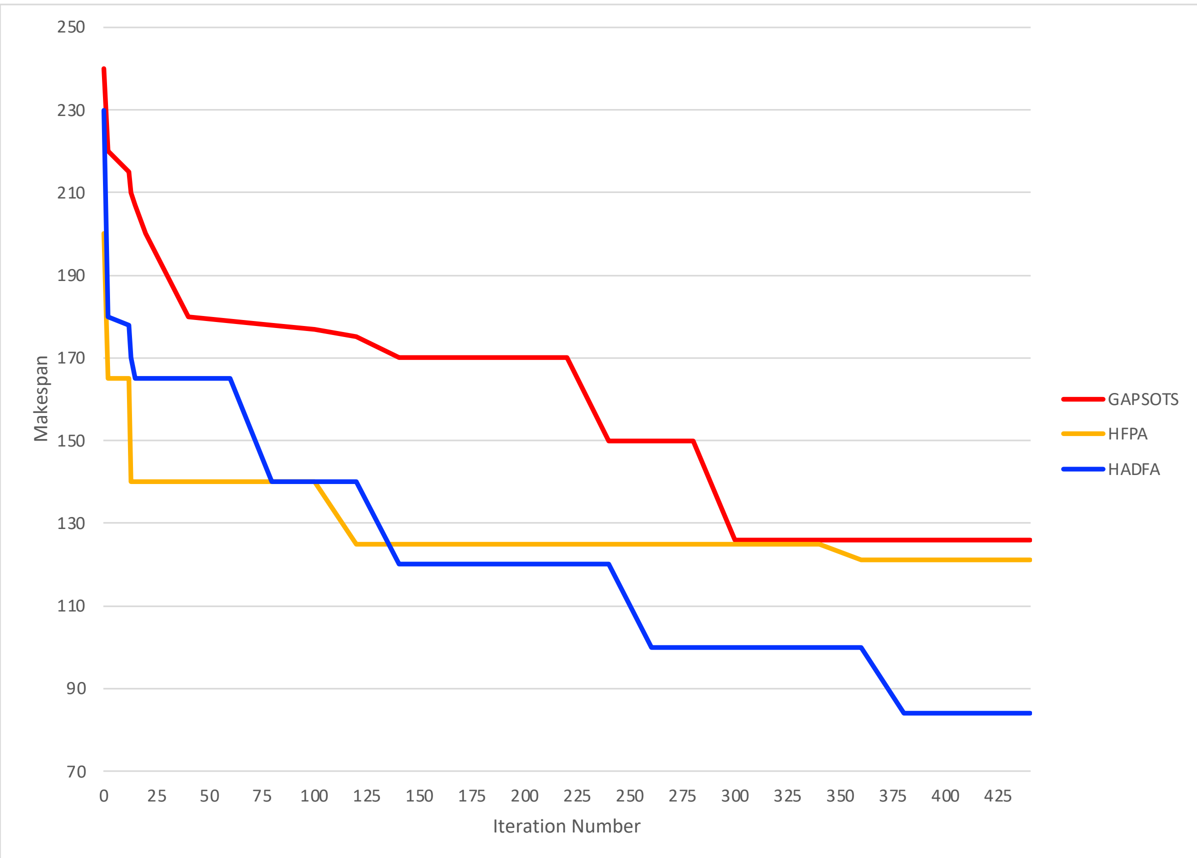


Figure 6.14 Convergence Comparison of Proposed Algorithms

6.8 SUMMARY:

Simultaneous scheduling of Machines and AGVs is carried out for single objective and multi objective. Minimization of makespan is considered as single objective and was solved by GAPSOTS, HFPA AND HAdFA for 82 problem sets proposed by Bilge and Ulusoy for two cases viz. $t/p > 0.25$; $t/p < 0.25$. A detailed analysis was performed and it is demonstrated that HAdFA outperforms GAPSOTS, HFPA and other existing algorithms. For multi objective optimisation, mean flow time and mean tardiness were considered in addition to makespan. They were tested for selective five test instances. For multi objective optimization also HAdFA proved to be more efficient in achieving best output results in comparison to other algorithms. The problem can be further expanded by assuming that each tool type has a single copy that is shared among all the machines and kept in a central tool magazine.

CHAPTER 7
MULTI OBJECTIVE SCHEDULING OF A REAL-LIFE CASE STUDY

7.1 INTRODUCTION

This chapter discusses a case study that addresses the problem of Partial-flexible job scheduling in an Lube Oil Blending Plant. A lubricating oil blending plant is a facility designed to produce a variety of lubricating oils by blending various base oils and additives according to specific formulations. Lubricating oils are essential for the proper functioning of machinery and equipment in a range of industries, including automotive, manufacturing, and aviation. The lubricants used in these industries must be of the highest quality to ensure optimal performance and minimize wear and tear on equipment. The blending process involves mixing the base oils with carefully selected additives, such as detergents, anti-wear agents, and viscosity modifiers, to produce lubricants that meet the specific needs of each application. The blending process must be carefully controlled to ensure consistency and quality, and the final products are subjected to rigorous testing to ensure they meet the required specifications. A lube oil blending plant is a critical part of the lubricants industry, playing a key role in the production of high-quality lubricating oils that keep machinery and equipment running smoothly and efficiently.

Lube oil blending can be identified as a partial flexible job shop problem as some of the machines may be capable of performing multiple operations or that some jobs may have different routes through the blending process. Because of this, the problem becomes dynamic and complex in nature. The blending process involves multiple operations that need to be performed in a particular sequence and with specific equipment and personnel requirements. The production process can be influenced by various factors such as the type of base oil, the specific additives required for a particular lubricant, and the specific order and time constraints for the production of different products. Moreover, production demand can be highly volatile, meaning that the plant needs to be able to quickly adjust production plans and schedules to accommodate changes in demand.

The flexible job shop problem is a well-known scheduling problem that involves scheduling jobs on various machines in a flexible manufacturing system. It is characterized by a high degree of variability in the production process, making it challenging to optimize production schedules and meet production targets. In the case of lube oil blending, the blending process is highly flexible, as the order of operations and the specific requirements can vary significantly between different products. Moreover, there are many constraints that need to be taken into account, such as the availability of specific equipment and personnel, the time required to clean equipment between runs, and the need to adhere to specific quality control standards.

To address the partial-FJSSP of Lube oil blending process, problem instances are taken from a range of 4 jobs x 5 machines and 10 jobs x 10 machines and are tested and evaluated for the performance and validity of proposed methods. Two proposed methods viz GAPSOTS and HAdFA are used in this study to minimize three objectives of minimizing makespan, total workload and machine workload.

7.2 STEPS IN LUBE OIL BLENDING PROCESS

In lubricant manufacturing, there are ever-changing complex and exacting formation requirements, whether material is processed in an in-line blending system or as a batch process. This requirement demands a control and information system that allows great flexibility in making constant changes to formulas and procedures, as well as being highly reliable and accurate. Various steps are involved in lube oil blending which are described in subsequent sections.

7.2.1 Base oil and additives selection:

The first step in lube oil blending production is to select the appropriate base oil or combination of base oils, depending on the desired properties of the finished lubricant. The base oils are typically stored in bulk tanks and are selected based on their viscosity, volatility, and chemical composition. Next additives are added. Additives are chemicals added to the base oil to enhance its performance, such as detergents, dispersants, viscosity modifiers, and anti-wear agents.

7.2.2 Blending

Blending is physical mixing of two or more components that are miscible in one another or made miscible to give a homogeneous mixture or solution. This is a purely Physical process. There is no change in chemical properties of components used in blending expect soluble cutting oil/coolant. Blending is a mechanical operation process involving the mixing different components in the correct proportion under the specified conditions, The product is formulated in the laboratory to meet the end-user requirements after extensive evaluation in the laboratory, test benches and field trials. The product is then characterized by laying down the specification and the control test limits.

For obtaining a homogeneous mixture or blend a certain sequence of charging of various components is followed in blending process, as per formulations. Charging sequence/order of each ingredient/component during blending is important for mutual compatibility/solubility of components. Any abrupt change in order/sequence may result to disruption of mutual of solubility of the components and which may further result separation of additives. Hence, in blending it is very essential to follow the recommended sequence of component while charging under specified condition as mentioned in BOM/formulations.

7.2.2.1 Important Factors in Blending

I. Cleanliness of kettle/blending sequence

II. Component charging sequence

III. Homogenization of all Components (Mixing/stirring)

IV. Dry Bases oil is another importance aspect in blending. If wet base oils containing water are used for blending, the result blend may become hazy any impossible to rectify due to interaction of the inherent water some of the additives forming finely divided insoluble particles in suspension.

V. Temperature of blending is yet another important parameter in manufacturing of lubricating oils, this is particularly important in blending of products with Group II/III base oils, soluble cutting oils, spray oils other specialty products.

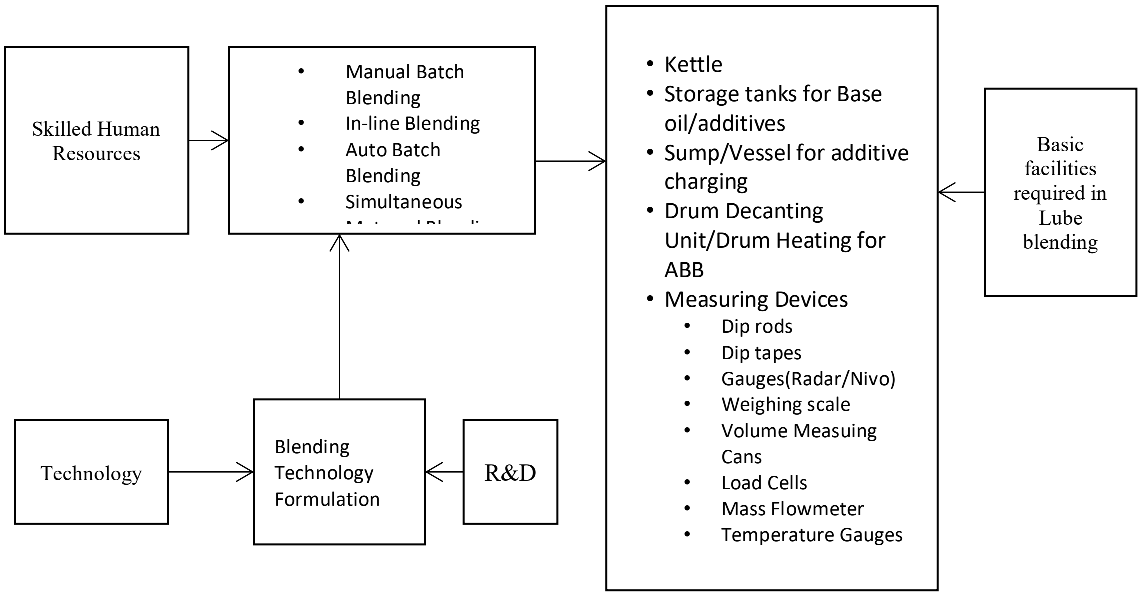


Figure 7.1 Flow diagram of blending process

Blending technologies

Blending may be carried using formulations/Bill of Material (BOM), which Have optimized

I. Volume (Components are Charged manually/auto in Volume unit)

II. Weight (Components are charged manually/auto in weight unit)

Blending is generally done by adopting one of the below mentioned technologies:

Manual or Conventional Batch Blending

Blending is carried out in kettles and all the operations are manually controlled. The charging of input i.e. base oils & additives may be partly automated in manual blending. Measurement of inputs either in volume or party in volume and party in weight (in case of solid additive).

Auto Batch Blending (ABB) System

ABB incorporates a primary blend vessel or tank equipped with a dynamic mixer, and one or dosing vessels mounted on load cells for measurement of liquid volume (or mass by ratio). Typically, all base oils and bulk additives are connected to the roof of the mixing kettle by dedicated piping to avoided contamination and accommodate automatic operation with minimal manual labour.

Based on precise weighing technology, the Automatic Batch Blending (ABB) System provides an efficient means of producing small volume blends. Generally designed to accommodate 1KL to 20KL blends when load cell weighing is employed, the ABB can support larger blends with mass flow meter-based systems. This plant has five ABB systems. Figure

7.2 shows the Auto Batch Blending (ABB) System. Figure 7.2 shows the manufacturing process of Lube oil Blending.

Figure 7.2 Auto Batch Blending (ABB) System

7.2.3 Quality control:

After the blending process, the lubricant is subjected to rigorous quality control testing to ensure that it meets the required specifications. The tests can include viscosity, pour point, flash point, and other physical and chemical properties.

7.2.4 Packaging:

After passing the quality control tests, the lubricant is packaged into containers of various sizes, such as drums, totes, or bulk tanks, depending on the customer's requirements.

7.2.5 Labeling and storage:

Finally, the lubricant is labeled with the appropriate information, including the product name, manufacturer, and specifications, and stored in a safe and secure location until it is shipped to the customer.

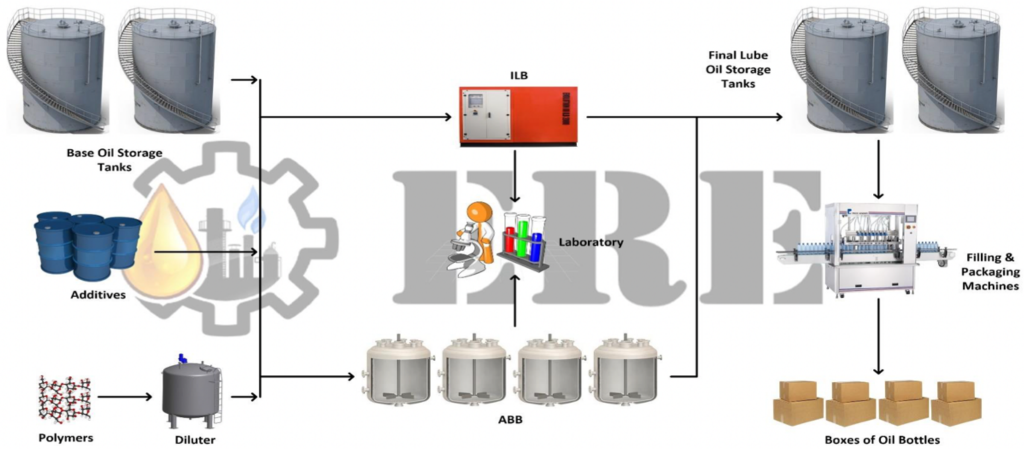


Figure 7.3 Manufacturing process of Lube oil Blending

7.3 INPUT DATA

Here the jobs are the products i.e the different lube oils manufactured like SERVO-spindle oils, Textile oils, Knitting oil, Automotive lubricating oils and Greases. Machines are different components for blending like Charging units of base oils, Mixing (ABB units), Discharge, Rinsing, Final mixing etc

Model 1 [4x5]

Initially, a small-scale instance that is 4 - machines, 5 - jobs is used to show the ability to optimize and test the hybrid algorithm's efficiency. Where the table represent the 4 machines x 5 Jobs with the variable time units. Let illustrate briefly about the table, the machine 1 carries five jobs, likewise, machine 2, 3 and 4 also carries 5 jobs. The variable time units are listed in the table, for example, in the given time period, machine 1 can process job 1 in 60 minutes followed by job 2 in 5 minutes, job 3 in 5 minutes, job 4 in 4 minutes and job 5 in 90 minutes. Similarly, for the machine 2 to 4, process is executed in the same way as machine 1.

Table 7.1 Input Data 1 from Oil Blending Plant [4- machines, 5- jobs]

Work Piece	**Job 1**	**Job 2**	**Job 3**	**Job4**	**Job5**
Machine 1	60	5	5	4	90
Machine 2	8	2	51	16	10
Machine 3	5	2	8	32	15
Machine 4	23	4	4	45	49

Model 2 [10x10]

In the case study 2, the small-scale instance that is 10 - machines, 10 - jobs is used to show the ability to optimize and test the hybrid algorithm's efficiency. Where the table represent the 10 machines x 10 Jobs with the variable time units. Let illustrate briefly about the table, the machine 1 carries ten jobs, likewise, machine 2, 3, 4, 5, 6, 7, 8, 9 and 10 also carries 10 jobs. The variable time units are listed in the table, for example, in the given time period, machine 1 can process job 1 in 60 minutes followed by job 2 in 5 minutes, job 3 in 5 minutes, job 4 in 4 minutes, job 5 in 90 minutes, job 6 in 60 minutes, job 7 in 26 minutes job 8 in 5 minutes, job 9 in 5 minutes and job 34 in 4 minutes. Similarly, for machine 2 to 10, process is executed in the same way as machine 1.

Table 7.2 Input Data 2 (Oil Blending Plant) [10- machines, 10- jobs]

Work Piece	Job 1	Job 2	Job 3	Job 4	Job 5	Job 6	Job 7	Job 8	Job 9	Job 10
Machine 1	60	5	5	4	90	60	26	5	5	34
Machine 2	8	2	51	16	10	30	35	8	5	23
Machine 3	5	2	8	32	15	67	34	3	8	65
Machine 4	23	4	4	45	49	41	35	1	17	67
Machine 5	4	4	5	24	90	52	65	2	18	43
Machine 6	6	8	4	34	45	46	13	3	31	27
Machine 7	8	6	51	22	10	53	21	6	5	43
Machine 8	23	2	8	12	15	35	34	6	6	24
Machine 9	23	24	45	34	49	35	32	7	6	31
Machine 10	2	8	6	34	5	50	13	8	4	28

7.4 RESULTS AND DISCUSSIONS

7.4.1 Model 1 Results:

Table 7.1 represent the problem and processing time for 4 Machines x 5 jobs. The machine assignment vector and operation scheduling vector of a solution is done by two string encoding method. The solution for the Model 1 is represented by a vector {3 1 4 1 2}. The second part is the scheduling component that indicates the scheduling sequence {5 3 1 2 4}. When a method of optimization is decoded, the operation scheduling vector is initially converted into an operation sequence. Then each operation is allocated to the processing machine to machine assignment vector. The Gantt chart for the decoded solution is illustrated in Figure 7.4 and 7.5 for HAdFA alone as it gives better results than GAPSOTS.

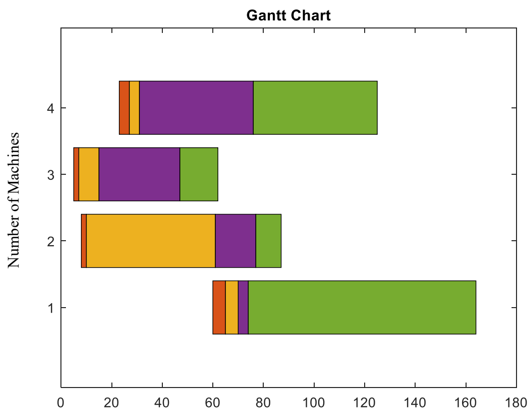


Figure 7.4 Gantt chart for 4x5

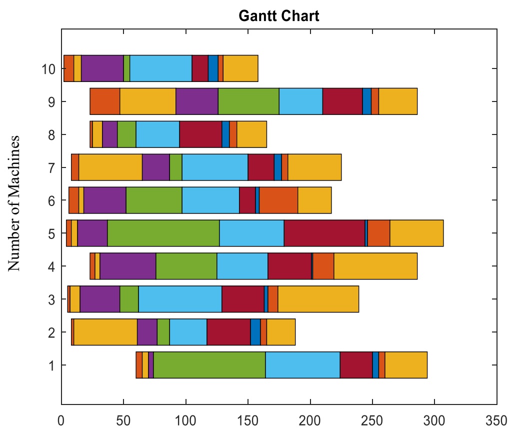


Figure 7.5 Gantt chart for 10x10

Model 2 Results:

Table 7.2 represent the problem and processing time for 10 Machines x 10 jobs. The machine assignment vector and operation scheduling vector of a solution for the given problem is done two string encoding method. The Gantt chart for the decoded solution is shown in Figure 7.5. The performance metrics obtained are shown in Table 7.3

Table 7.3 Obtained results using HAdFA for oil blending plant

	4x5		10x10	
	GAPSOTS	HADFA	GAPSOTS	HADFA
Makespan	8	6	24	22
Maximal machine workload	36	33	145	146
Total workload of machines	464	455	815	811
COF	12.24	10.91	11.468	9.23
Computational time (s)	0.990515 seconds	0.678546 seconds	1.130247 seconds	0.8843768 seconds

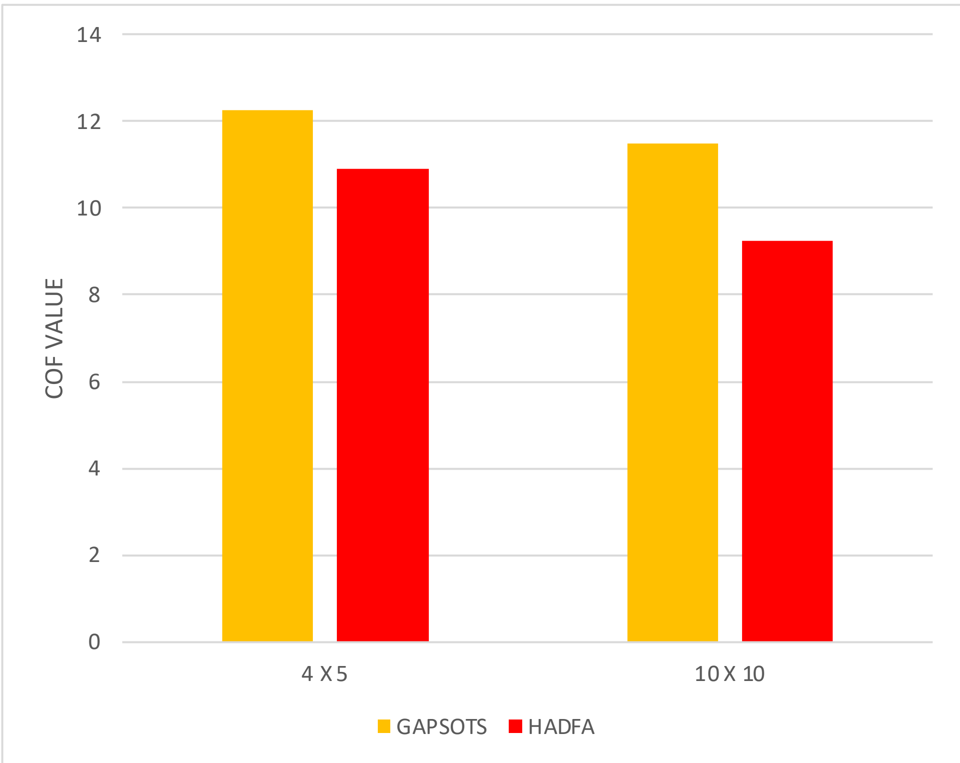


Figure 7.6 COF comparison for proposed algorithms

Figure 7.6 shows the COF comparison of the proposed algorithms GAPSOTS and HADFA. It can be inferred from Table 7.3 that HADFA tends to give better results than GAPSOTS.

In all types of problems studied in this research, HADFA is consistently performing better thereby demonstrating that HADFA is the best algorithm in terms of computational wise and optimal results wise. The total time to perform the optimization is mere seconds proving it has high efficiency than GAPSOTS.

7.5 SUMMARY

This chapter discussed about the implementation of GAPSOTS and HADFA for a real-life case study conducted in a Lube Oil Blending plant in Faridabad. The problem type identified is partial-FJSSP. HADFA gave better performance than GAPSOTS for obtaining COF for combined objectives of makespan, total workload and maximum workload.

CHAPTER 8
CONCLUSIONS AND FUTURE SCOPE

8.1 THESIS OUTCOME

This thesis addresses the scheduling of various resources in an FMS environment for the minimization of various objectives in order to increase productivity and lower manufacturing costs. The proposed approaches have been tested using numerical problems of various sizes as well as benchmark problems. To verify the effectiveness of the suggested approaches, the results are compared and validated with those of alternative algorithms present in literature. Different objectives have been taken as single and multiple objectives for minimization. For multiple objectives, they are all taken into account combinedly. The entire research has been done in three parts.

1. Flexible Job shop Scheduling Problem- A multi-objective approach
2. Combined Objective Optimization of Flexible Manufacturing System Scheduling
3. Concurrent scheduling of machines and AGVs.
4. Application of Proposed Algorithm on a Real-Life case study.

The contributions of the research scholar for this thesis include:

- Identify the research problem in Flexible Manufacturing System
- A thorough study of books and research articles that discuss the literature that is relevant to this work.
- Gathering the necessary information from industry-related and cited literature studies for the improvement of FMS Scheduling.
- HAdFA, GAPSOTS, and HFPA were chosen as the most appropriate Metaheuristic optimization strategies from the available techniques for this work to solve the identified research problem.
- Developed the aforementioned algorithms and implemented for the Scheduling of FMS for a set of objectives.
- Programmed the code for proposed algorithms in MATLAB 2019a.
- Performed Optimization of FMS scheduling for single objective and multi-objectives.
- Striking an effective analogy among three nature inspired algorithms, viz., HAdFA, GAPSOTS, and HFPA and the problems and processes of FMS to achieve optimal solution in minimum possible time so as to increase the production efficiency.

The conclusions derived from each part of research problems are discussed in subsequent sections.

8.2 CONCLUSIONS FOR FJSSP

- The multi-objective functions of minimizing makespan, total workload, maximum workload, total idle time, and total tardiness were considered as objectives to solve FJSSP .
- The proposed HAdFA employs an adaptive strategy, where the parameters are dynamically updated during the iterations thereby striking a balance between exploration and exploitation leading to get new optimal solutions.
- HAdFA and HFPA were tested on benchmark problems of Kacem, DP data, BR data, and Du data for several problems of different sizes for FJSSP.
- Due to the adaptive strategies used in our approach, **new Pareto optimal solutions** MSmax = 10, WLmax = 8, and WLtotal = 32, and MSmax = 12, WLmax = 9, and WLtotal = 34 were found for three problems in Test instance 1 (Kacem data).
- In Test Instance 2 (DP data), **new makespan (MSmax)** values were obtained for all problems except five problems.
- Best makespan values are obtained for large scale problems for BR data by HAdFA especially MK10 has new makespan value of **20.1**
- The convergence graph analysis shows that HAdFA converges faster and obtain optimal makespan in less computational time.

8.3 CONCLUSIONS FOR COMBINED OBJECTIVE FUNCTION FOR FMS

- Four benchmark problems of varying jobs and machines for a particular FMS set up has been optimised.
- The proposed algorithms GAPSOTS, HAdFA and HFPA are implemented to solve Combined Objective Function (COF) which minimizes the total idle time of a machine and total penalty cost.
- It is found that HAdFA gave better COF when compared to HFPA and GAPSOTS. Nevertheless the other algorithms also gave near optimal solutions.
- The results demonstrates that the methods created here may be effectively customized to work with any FMS that has numerous components and machines and can perform optimization for any mix of jobs and machines that are subject to multiple objective functions.

8.4 CONCLUSIONS FOR CONCURRENT SCHEDULING OF MACHINES AND AUTOMATIC GUIDED VEHICLES

- Concurrent scheduling of Machines and AGVs is carried out for single objective and multi objective. Minimization of makespan is considered as single objective and was solved by GAPSOTS, HFPA AND HAdFA for 82 problem sets proposed by Bilge and Ulusoy for two cases viz. $t/p > 0.25$; $t/p < 0.25$.
- For multi objective optimisation, mean flow time and mean tardiness were considered in addition to makespan. They were tested for selective five test instances.
- For simultaneous scheduling of machines and AGVs, improvement in scheduling is possible by HAdFA, particularly in settings where cycle durations are quick and trip times are comparable, or if the layout and the process routes don't work well together.
- The outcomes confirms that HAdFA frequently produces superior results than other algorithms and conventional techniques.
- A convergence analysis of proposed algorithms proves that HAdFA can perform better and converge better.

8.5 CONCLUSIONS FOR REAL LIFE CASE STUDY

- A real-life case study was performed in Lube Oil Plant in Faridabad. Lube oil blending production can be identified as a partial flexible job shop problem as some of the machines may be capable of performing multiple operations or that some jobs may have different routes through the blending process. Because of this, the problem becomes dynamic and complex in nature
- The proposed algorithms GAPSOTS and HADFA are implemented for the identified problem. The problem is to optimize multi objectives of minimization of total makespan, workload and maximum workload is performed.
- HADFA outperformed GAPSOTS and gave minimum makespan and COF. The computational time taken by HADFA is significantly less than GAPSOTS. For real time production these few seconds computational capacity matters most as it results in high efficiency of running a plant.

8.6 LIMITATIONS OF THIS WORK

- Multi-objective optimization has only taken time-based objectives into account.
- The tested data for single objective optimization, multi objective optimization and total holding cost optimization for AGV scheduling are benchmark problems.
- Machine break down time, Tool set up time, Tool transfers are not considered. In future, the problems can be considered with these criteria.

8.7 FUTURE SCOPE OF THE PRESENT WORK

The future scope of Flexible Manufacturing System (FMS) scheduling is vast and diverse. Some of the potential areas for future research and development in this field include:

- Integration with Industry 5.0 technologies: The adoption of Industry 5.0 technologies, such as the Internet of Things (IoT), cloud computing, and artificial intelligence (AI), can enable more efficient and intelligent FMS scheduling. For example, real-time data from machines and sensors can be used to optimize scheduling decisions, and AI algorithms can be used to automatically generate optimal schedules.
- Dynamic scheduling: FMS scheduling is often performed offline, based on a predetermined set of jobs and operations. However, in dynamic environments where new orders and changes occur frequently, dynamic scheduling techniques that can adapt to changing conditions in real-time can be highly beneficial.
- Distributed and decentralized scheduling: In FMS environments with multiple machines and resources, centralized scheduling may not be practical or optimal. Future research can focus on developing distributed and decentralized scheduling algorithms that can coordinate the scheduling decisions of multiple agents and resources.
- Green manufacturing: With increasing concerns about environmental sustainability, future research can focus on developing FMS scheduling techniques that can minimize energy consumption, reduce waste, and promote eco-friendly production.
- Overall, the future scope of FMS scheduling is vast and evolving, with many potential areas for research and development that can lead to more efficient, intelligent, and sustainable manufacturing systems.

REFERENCES

[1] R. Kumar, M. K. Tiwari, and R. Shankar, “Scheduling of flexible manufacturing systems: An ant colony optimization approach,” *Proc Inst Mech Eng B J Eng Manuf*, vol. 217, no. 10, pp. 1443–1453, Oct. 2003, doi: 10.1243/095440503322617216.

[2] A. K. Dhingra, “Multi-objective SDST flow shop scheduling using Genetic Algorithm : An industrial Case study,” *Int J Latest Trends Eng Technol*, vol. 5, no. 2, pp. 465–468, 2015.

[3] N. Rajan, S. Jaiswal, T. Kalsi, and V. Singh, “Scheduling of Flexible Manufacturing System using Genetic Algorithm (Multiobjective): A Review,” *Int J Comput Appl*, vol. 86, no. 19, pp. 9–15, Jan. 2014, doi: 10.5120/15102-2678.

[4] J. Browne, D. Dubois, K. Rathmill, S. Sethi, and K. Stecke, *Classification of Flexible Manufacturing Systems*, vol. 2. 1984.

[5] K. Amoako-Gyampah and M. J. Maffei, “The adoption of flexible manufacturing systems: strategic considerations,” *Technovarion*, vol. 9, pp. 479–491, 1989.

[6] K. P. Sahu and R. Tamralkar, “Flexible Manufacturing System,” *National Conference on Recent Trends in Renewable Energy Sources & Electronics*, vol. 3, no. 2, pp. 30–35, 2016.

[7] A. Kaushal, A. Vardhan, and Rajput, “Flexible Manufacturing System A Modern Approach To Manufacturing Technology,” *International Refereed Journal of Engineering and Science*, vol. 5, no. 4, pp. 16–23, 2016.

[8] J. C. Ammons, T. Govindaraj, and C. M. Mitchell, “A supervisory control paradigm for real-time control of flexible manufacturing systems,” *Ann Oper Res*, vol. 15, no. 1, pp. 313–335, Dec. 1988, doi: 10.1007/BF02186801.

[9] P. Udhayakumar, Guide, and S. Kumanan, “Scheduling of flexible manufacturing system using metaheuristics,” Anna University, 2011.

[10] A. R. Moro, H. Yu, and G. Kelleher, “Advanced scheduling methodologies for flexible manufacturing systems using Petri nets and heuristic search,” in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, 2000, vol. 3, pp. 2398–2403. doi: 10.1109/ROBOT.2000.846386.

[11] C. Saygin, S. E. Kiliç, T. Tóth, and F. Erdélyi, “On Scheduling Approaches of Flexible Manufacturing Systems: Gap Between Theory and Practice,” *IFAC Proceedings Volumes*, vol. 28, no. 24, pp. 61–66, Oct. 1995, doi: 10.1016/S1474-6670(17)46526-8.

[12] M. Yadollahi and A. M. Rahmani, “Solving Distributed Flexible Manufacturing Systems Scheduling Problems Subject to Maintenance: Memetic Algorithms Approach,” in *2009 Ninth IEEE International Conference on Computer and Information Technology*, 2009, pp. 36–41. doi: 10.1109/CIT.2009.113.

[13] G. Chryssolouris, J. Pierce, and K. Dicke, “A decision-making approach to the operation of flexible manufacturing systems,” *International Journal of Flexible Manufacturing Systems*, vol. 4, no. 3–4, pp. 309–330, Jun. 1992, doi: 10.1007/BF01324885.

[14] R. Ingalls, “Introduction to Simulation,” in *Winter Simulation Conference*, 2008, pp. 1–10.

[15] A. Yadav and S. C. Jayswal, “Makespan Minimization in Flexible Manufacturing System using Genetic Algorithm,” *International Journal of Applied Engineering Research*, vol. 13, no. 11, pp. 9328–9334, 2018.

[16] 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*, vol. 3, pp. 1740–1754, 2015, doi: 10.1109/ACCESS.2015.2481463.

[17] M. Muthukumaran and S. Muthu, "A Heuristic Scheduling Algorithm for Minimizing Makespan and Idle Time in a Nagare Cell," *Advances in Mechanical Engineering*, vol. 4, p. 895463, Jan. 2015, doi: 10.1155/2012/895463.

[18] V. K. Chawla, A. K. Chanda, and S. Angra, "Sustainable multi-objective scheduling for automatic guided vehicle and flexible manufacturing system by a grey wolf optimization algorithm," *International Journal of Data and Network Science*, pp. 27–40, 2018, doi: 10.5267/j.ijdns.2018.6.001.

[19] E. Shokouhi, "Integrated multi-objective process planning and flexible job shop scheduling considering precedence constraints," *Prod Manuf Res*, vol. 6, no. 1, pp. 61–89, Jan. 2017, doi: 10.1080/21693277.2017.1415173.

[20] B. Veeravalli, G. Rajesh, and N. Viswanadham, "Design and analysis of optimal material distribution policies in flexible manufacturing systems using a single AGV," *Int J Prod Res*, vol. 40, no. 12, pp. 2937–2954, Jan. 2002, doi: 10.1080/00207540210137648.

[21] J. Wang and Y. Deng, "Incremental modeling and verification of flexible manufacturing systems," *J Intell Manuf*, vol. 10, pp. 485–502, 1999.

[22] J. Jerald, P. Asokan, R. Saravanan, and A. D. C. Rani, "Simultaneous scheduling of parts and automated guided vehicles in an FMS environment using adaptive genetic algorithm," *The International Journal of Advanced Manufacturing Technology*, vol. 29, no. 5–6, pp. 584–589, Sep. 2006, doi: 10.1007/BF02729112.

[23] J. M. Novas and G. P. Henning, "Integrated scheduling of resource-constrained flexible manufacturing systems using constraint programming," *Expert Syst Appl*, vol. 41, no. 5, pp. 2286–2299, Apr. 2014, doi: 10.1016/j.eswa.2013.09.026.

[24] A. Allahverdi, J. N. D. Gupta, and T. Aldowaisan, "A review of scheduling research involving setup considerations," *Omega (Westport)*, vol. 27, no. 2, pp. 219–239, Apr. 1999, doi: 10.1016/S0305-0483(98)00042-5.

[25] W.-H. Yang and C. J. Liao, "Survey of scheduling research involving setup times," *Int J Syst Sci*, vol. 30, no. 2, pp. 143–155, Jan. 1999, doi: 10.1080/002077299292498.

[26] T. C. E. Cheng, J. N. D. Gupta, and G. Wang, "A Review of Flowshop Scheduling Research with Setup Times," *Prod Oper Manag*, vol. 9, no. 3, pp. 262–282, Jan. 2000, doi: 10.1111/j.1937-5956.2000.tb00137.x.

[27] C. N. Potts and M. Y. Kovalyov, "Scheduling with batching: A review," *Eur J Oper Res*, vol. 120, no. 2, pp. 228–249, Jan. 2000, doi: 10.1016/S0377-2217(99)00153-8.

[28] W.-H. Yang and C. J. Liao, "Survey of scheduling research involving setup times," *Int J Syst Sci*, vol. 30, no. 2, pp. 143–155, Jan. 1999, doi: 10.1080/002077299292498.

[29] T. C. E. Cheng, J. N. D. Gupta, and G. Wang, "A Review of Flowshop Scheduling Research with Setup Times," *Prod Oper Manag*, vol. 9, no. 3, pp. 262–282, Jan. 2000, doi: 10.1111/j.1937-5956.2000.tb00137.x.

[30] C. N. Potts and M. Y. Kovalyov, "Scheduling with batching: A review," *Eur J Oper Res*, vol. 120, no. 2, pp. 228–249, Jan. 2000, doi: 10.1016/S0377-2217(99)00153-8.

[31] A. Allahverdi, J. N. D. Gupta, and T. Aldowaisan, "A review of scheduling research involving setup considerations," *Omega (Westport)*, vol. 27, no. 2, pp. 219–239, Apr. 1999, doi: 10.1016/S0305-0483(98)00042-5.

[32] A. Allahverdi, C. T. Ng, T. C. E. Cheng, and M. Y. Kovalyov, "A survey of scheduling problems with setup times or costs," *Eur J Oper Res*, vol. 187, no. 3, pp. 985–1032, Jun. 2008, doi: 10.1016/j.ejor.2006.06.060.

[33] B. Veeravalli, G. Rajesh, and N. Viswanadham, “Design and analysis of optimal material distribution policies in flexible manufacturing systems using a single AGV,” *Int J Prod Res*, vol. 40, no. 12, pp. 2937–2954, Jan. 2002, doi: 10.1080/00207540210137648.

[34] J. Wang and Y. Deng, “Incremental modeling and verification of flexible manufacturing systems,” *J Intell Manuf*, vol. 10, pp. 485–502, 1999.

[35] J. Jerald, P. Asokan, R. Saravanan, and A. D. C. Rani, “Simultaneous scheduling of parts and automated guided vehicles in an FMS environment using adaptive genetic algorithm,” *The International Journal of Advanced Manufacturing Technology*, vol. 29, no. 5–6, pp. 584–589, Sep. 2006, doi: 10.1007/BF02729112.

[36] J. M. Novas and G. P. Henning, “Integrated scheduling of resource-constrained flexible manufacturing systems using constraint programming,” *Expert Syst Appl*, vol. 41, no. 5, pp. 2286–2299, Apr. 2014, doi: 10.1016/j.eswa.2013.09.026.

[37] A. Allahverdi, C. T. Ng, T. C. E. Cheng, and M. Y. Kovalyov, “A survey of scheduling problems with setup times or costs,” *Eur J Oper Res*, vol. 187, no. 3, pp. 985–1032, Jun. 2008, doi: 10.1016/j.ejor.2006.06.060.

[38] T. Sawik, “A multilevel machine and vehicle scheduling in a flexible manufacturing system,” *Math Comput Model*, vol. 23, no. 7, pp. 45–57, Apr. 1996, doi: 10.1016/0895-7177(96)00028-3.

[39] G. Ulusoy and Ü. Bilge, “Simultaneous scheduling of machines and automated guided vehicles,” *Int J Prod Res*, vol. 31, no. 12, pp. 2857–2873, Dec. 1993, doi: 10.1080/00207549308956904.

[40] Ü. Bilge and G. Ulusoy, “A Time Window Approach to Simultaneous Scheduling of Machines and Material Handling System in an FMS,” *Oper Res*, vol. 43, no. 6, pp. 1058–1070, Dec. 1995, doi: 10.1287/opre.43.6.1058.

[41] J. Blazewicz, H. A. Eiselt, G. Finke, G. Laporte, and J. Weglarz, “Scheduling tasks and vehicles in a flexible manufacturing system,” *International Journal of Flexible Manufacturing Systems*, vol. 4, no. 1, pp. 5–16, Aug. 1991, doi: 10.1007/BF01325094.

[42] Y. A. Bozer, “Guided Vehicle Systems: Information/Control System Implications of Alternative Design and Operation Strategies,” in *Advanced Information Technologies for Industrial Material Flow Systems*, Berlin, Heidelberg: Springer Berlin Heidelberg, 1989, pp. 417–433. doi: 10.1007/978-3-642-74575-1_19.

[43] R. Ram and N. Viswanadhamj, “Performability of automated manufacturing systems with centralized material handling,” *Int J Prod Res*, vol. 32, no. 8, pp. 1775–1799, Aug. 1994, doi: 10.1080/00207549408957041.

[44] M. F. Anwar and R. Nagi, “Integrated scheduling of material handling and manufacturing activities for just-in-time production of complex assemblies,” *Int J Prod Res*, vol. 36, no. 3, pp. 653–681, Mar. 1998, doi: 10.1080/002075498193624.

[45] Ü. Bilge and G. Ulusoy, “A Time Window Approach to Simultaneous Scheduling of Machines and Material Handling System in an FMS,” *Oper Res*, vol. 43, no. 6, pp. 1058–1070, Dec. 1995, doi: 10.1287/opre.43.6.1058.

[46] E. Ahmadi, M. Zandieh, M. Farrokh, and S. M. Emami, “A multi objective optimization approach for flexible job shop scheduling problem under random machine breakdown by evolutionary algorithms,” *Comput Oper Res*, vol. 73, pp. 56–66, Sep. 2016, doi: 10.1016/j.cor.2016.03.009.

[47] R. Kumar, M. K. Tiwari, and R. Shankar, “Scheduling of flexible manufacturing systems: An ant colony optimization approach,” *Proc Inst Mech Eng B J Eng Manuf*, vol. 217, no. 10, pp. 1443–1453, Oct. 2003, doi: 10.1243/095440503322617216.

[48] C. Basnet and J. H. Mize, "Scheduling and control of flexible manufacturing systems: a critical review," *Int J Comput Integr Manuf*, vol. 7, no. 6, pp. 340–355, Nov. 1994, doi: 10.1080/09511929408944622.

[49] A. Grieco, Q. Semeraro, and T. Tolio, "A Review of Different Approaches to the FMS Loading Problem," *International Journal of Flexible Manufacturing Systems*, vol. 13, no. 4, pp. 361–384, 2001, doi: 10.1023/A:1012290630540.

[50] M. Dios and J. M. Framinan, "A review and classification of computer-based manufacturing scheduling tools," *Comput Ind Eng*, vol. 99, pp. 229–249, Sep. 2016, doi: 10.1016/j.cie.2016.07.020.

[51] J. Blazewicz, H. A. Eiselt, G. Finke, G. Laporte, and J. Weglarz, "Scheduling tasks and vehicles in a flexible manufacturing system," *International Journal of Flexible Manufacturing Systems*, vol. 4, no. 1, pp. 5–16, Aug. 1991, doi: 10.1007/BF01325094.

[52] C. Pach, T. Berger, T. Bonte, and D. Trentesaux, "ORCA-FMS: a dynamic architecture for the optimized and reactive control of flexible manufacturing scheduling," *Comput Ind*, vol. 65, no. 4, pp. 706–720, May 2014, doi: 10.1016/j.compind.2014.02.005.

[53] M. Dios and J. M. Framinan, "A review and classification of computer-based manufacturing scheduling tools," *Comput Ind Eng*, vol. 99, pp. 229–249, Sep. 2016, doi: 10.1016/j.cie.2016.07.020.

[54] M. Gen, W. Zhang, L. Lin, and Y. Yun, "Recent advances in hybrid evolutionary algorithms for multiobjective manufacturing scheduling," *Comput Ind Eng*, Jan. 2017, doi: 10.1016/j.cie.2016.12.045.

[55] M. V. Magalhaes and N. Shah, "Crude Oil Scheduling," *Foundations of Computer-Aided Operations (FOCAPO)*, 2003.

[56] I. Simeonovova, A. Gyurov, and S. Simeonov, "Modern Methods for Manufacturing Planning and Scheduling," *MM Science Journal*, vol. 2015, no. 03, pp. 635–639, Sep. 2015, doi: 10.17973/MMSJ.2015_10_201519.

[57] K. E. Stecke, "Design, planning, scheduling, and control problems of flexible manufacturing systems," *Ann Oper Res*, vol. 3, no. 1, pp. 1–12, 1985, doi: 10.1007/BF02023765.

[58] R. A. Liaqait, S. Hamid, S. S. Warsi, and A. Khalid, "A critical analysis of job shop scheduling in context of Industry 4.0," *Sustainability (Switzerland)*, vol. 13, no. 14, pp. 1–19, 2021, doi: 10.3390/su13147684.

[59] C. Basnet and J. H. Mize, "Scheduling and control of flexible manufacturing systems: a critical review," *Int J Comput Integr Manuf*, vol. 7, no. 6, pp. 340–355, 1994, doi: 10.1080/09511929408944622.

[60] R. Ojstersek, M. Brezocnik, and B. Buchmeister, "Multi-objective optimization of production scheduling with evolutionary computation: A review," *International Journal of Industrial Engineering Computations*, vol. 11, no. 3, pp. 359–376, 2020, doi: 10.5267/j.ijiec.2020.1.003.

[61] Y. He, K. E. Stecke, and M. L. Smith, "Robot and machine scheduling with state-dependent part input sequencing in flexible manufacturing systems," *Int J Prod Res*, vol. 54, no. 22, pp. 6736–6746, 2016, doi: 10.1080/00207543.2016.1161252.

[62] A. Negahban and J. S. Smith, "Simulation for manufacturing system design and operation: Literature review and analysis," *J Manuf Syst*, vol. 33, no. 2, pp. 241–261, 2014, doi: 10.1016/j.jmsy.2013.12.007.

[63] M. Freitag and T. Hildebrandt, “Automatic design of scheduling rules for complex manufacturing systems by multi-objective simulation-based optimization,” *CIRP Ann Manuf Technol*, vol. 65, no. 1, pp. 433–436, 2016, doi: 10.1016/j.cirp.2016.04.066.

[64] L. S. R. Krishna, T. S. S. Saketh, and M. V. S. Kumar, “A Simulation Based Approach for Studying the Effect of Buffers on the Performance of an FMS,” *International Journal of Engineering Science Invention*, vol. 4, no. 1, pp. 5–9, 2015.

[65] X. Gang and Q. Quan, “Research on Planning Scheduling of Flexible Manufacturing System Based on Multi-level List Algorithm,” *Procedia CIRP*, vol. 56, pp. 569–573, 2016, doi: 10.1016/j.procir.2016.10.112.

[66] K. Mallikarjuna, K. Hema Chandra Reddy, N. Govinda Rao, and V. Veeranna, “Integrated Scheduling of Ladder layout design in Flexible Manufacturing System using Non-Traditional optimization techniques,” *Mater Today Proc*, vol. 4, no. 8, pp. 8069–8078, 2017, doi: 10.1016/j.matpr.2017.07.146.

[67] M. E. Erdin and A. Atmaca, “Implementation of an Overall Design of a Flexible Manufacturing System,” *Procedia Technology*, vol. 19, pp. 185–192, 2015, doi: 10.1016/j.protcy.2015.02.027.

[68] Y.-H. Liu, H.-P. Huang, and Yu-Sheng Lin, “Dynamic scheduling of flexible manufacturing system using support vector machines,” in *IEEE International Conference on Automation Science and Engineering, 2005.*, 2005, pp. 387–392. doi: 10.1109/COASE.2005.1506800.

[69] H. C. Lee and C. Ha, “Sustainable integrated process planning and scheduling optimization using a genetic algorithm with an integrated chromosome representation,” *Sustainability (Switzerland)*, vol. 11, no. 2, 2019, doi: 10.3390/su11020502.

[70] R. Malik and P. N. Pena, “Optimal Task Scheduling in a Flexible Manufacturing System using Model Checking,” *IFAC-PapersOnLine*, vol. 51, no. 7, pp. 230–235, 2018, doi: 10.1016/j.ifacol.2018.06.306.

[71] O. T. Baruwa and M. A. Piera, “Identifying FMS repetitive patterns for efficient search-based scheduling algorithm: A colored Petri net approach,” *J Manuf Syst*, vol. 35, pp. 120–135, Apr. 2015, doi: 10.1016/j.jmsy.2014.11.009.

[72] J. Lee and J. S. Lee, “Heuristic search for scheduling flexible manufacturing systems using lower bound reachability matrix,” *Comput Ind Eng*, vol. 59, no. 4, pp. 799–806, Nov. 2010, doi: 10.1016/j.cie.2010.08.006.

[73] H. Yu, A. Reyes, S. Cang, and S. Lloyd, “Combined Petri net modelling and AI based heuristic hybrid search for flexible manufacturing systems—part 1. Petri net modelling and heuristic search,” *Comput Ind Eng*, vol. 44, no. 4, pp. 527–543, Apr. 2003, doi: 10.1016/S0360-8352(02)00212-7.

[74] G. Zambrano Rey, T. Bonte, V. Prabhu, and D. Trentesaux, “Reducing myopic behavior in FMS control: A semi-heterarchical simulation–optimization approach,” *Simul Model Pract Theory*, vol. 46, pp. 53–75, Aug. 2014, doi: 10.1016/j.simpat.2014.01.005.

[75] M. Mastrolilli and L. M. Gambardella, “Effective neighbourhood functions for the flexible job shop problem,” *Journal of Scheduling*, vol. 3, no. 1, pp. 3–20, 2000, doi: 10.1002/(sici)1099-1425(200001/02)3:1<3::aid-jos32>3.0.co;2-y.

[76] Y. Mati, N. Rezg, and X. Xie, “An integrated greedy heuristic for a flexible job shop scheduling problem,” *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 2534–2539, 2001, doi: 10.1109/ICSMC.2001.972939.

[77] W. Xia and Z. Wu, “An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems,” *Comput Ind Eng*, vol. 48, no. 2, pp. 409–425, Mar. 2005, doi: 10.1016/j.cie.2005.01.018.

[78] A. Motaghedi-larijani, K. Sabri-Laghaie, and M. Heydari, “Solving Flexible job shop scheduling with Multi Objective Approach,” *International Journal of Industrial Engineering & Production Research*, vol. 21, no. 4, pp. 197–209, 2010.

[79] L.-N. Xing, Y.-W. Chen, P. Wang, Q.-S. Zhao, and J. Xiong, “A Knowledge-Based Ant Colony Optimization for Flexible Job Shop Scheduling Problems,” *Appl Soft Comput*, vol. 10, pp. 888–896, 2010, doi: 10.1016/j.asoc.2009.10.006.

[80] A. Khadwilard, S. Chansombat, T. Thepphakorn, P. Thapatsuwan, W. Chainate, and P. Pongcharoen, “Application of firefly algorithm and its parameter setting for job shop scheduling,” *The Journal of Industrial Technology*, vol. 8, no. 1, 2012.

[81] V. Bhushan and M. Kumar, “Scheduling of Flexible Manufacturing System by Using Genetic Algorithm and Taguchi Philosophy,” *International Journal of Mechanical Engineering and Information Technology*, vol. 2, no. 6, pp. 550–555, 2014.

[82] S. Karthikeyan, P. Asokan, S. Nickolas, and T. Page, “Solving flexible job-shop scheduling problem using hybrid particle swarm optimisation algorithm and data mining,” *International Journal of Manufacturing Technology and Management*, vol. 26, no. 1–4, pp. 81–103, 2012, doi: 10.1504/IJMTM.2012.051445.

[83] V. Yannibelli and A. Amandi, “Hybridizing a multi-objective simulated annealing algorithm with a multi-objective evolutionary algorithm to solve a multi-objective project scheduling problem,” *Expert Syst Appl*, vol. 40, no. 7, pp. 2421–2434, 2013, doi: 10.1016/j.eswa.2012.10.058.

[84] V. Roshanaei, A. Azab, and H. Elmaraghy, “Mathematical modelling and a meta-heuristic for flexible job shop scheduling,” *Int J Prod Res*, vol. 51, no. 20, pp. 6247–6274, 2013, doi: 10.1080/00207543.2013.827806.

[85] Y. Yuan, H. Xu, and J. Yang, “A hybrid harmony search algorithm for the flexible job shop scheduling problem,” *Applied Soft Computing Journal*, vol. 13, no. 7, pp. 3259–3272, 2013, doi: 10.1016/j.asoc.2013.02.013.

[86] R. Buddala and S. S. Mahapatra, “An integrated approach for scheduling flexible job-shop using teaching–learning-based optimization method,” *Journal of Industrial Engineering International*, vol. 15, no. 1, pp. 181–192, 2019, doi: 10.1007/s40092-018-0280-8.

[87] Y. Li, J. Wang, and Z. Liu, “An Adaptive Multi-Objective Evolutionary Algorithm with Two-Stage Local Search for Flexible Job-Shop Scheduling,” *International Journal of Computational Intelligence Systems*, vol. 14, no. 1, pp. 54–66, 2020, doi: https://doi.org/10.2991/ijcis.d.201104.001.

[88] T. Ning, Z. Wang, X. Duan, and X. Liu, “Research on flexible job shop scheduling with low-carbon technology based on quantum bacterial foraging optimization,” *International Journal of Low-Carbon Technologies*, Jan. 2021, doi: 10.1093/ijlct/ctab005.

[89] T. Jiang, H. Zhu, J. Gu, L. Liu, and H. Song, “A discrete animal migration algorithm for dual-resource constrained energy-saving flexible job shop scheduling problem,” *Journal of Intelligent & Fuzzy Systems*, vol. 42, pp. 3431–3444, 2022, doi: 10.3233/JIFS-211399.

[90] X. Long, J. Zhang, K. Zhou, and T. Jin, "Dynamic Self-Learning Artificial Bee Colony Optimization Job Insertion," *Processes*, vol. 10, no. 571, 2022, doi: https://doi.org/10.3390/pr10030571.

[91] V. K. Chawla, A. K. Chanda, and S. Angra, "Scheduling of multi load AGVs in FMS by modified memetic particle swarm optimization algorithm," *Journal of Project Management*, vol. 3, pp. 39–54, 2018, doi: 10.5267/j.jpm.2017.10.001.

[92] N. S. Reddy, D. v. Ramamurthy, M. P. Lalitha, and K. P. Rao, "Minimizing the total completion time on a multi-machine FMS using flower pollination algorithm," *Soft comput*, vol. 26, no. 3, pp. 1437–1458, 2022, doi: 10.1007/s00500-021-06411-y.

[93] Y. Zheng, Y. Xiao, and Y. Seo, "A tabu search algorithm for simultaneous machine/AGV scheduling problem," *Int J Prod Res*, vol. 52, no. 19, pp. 5748–5763, 2014, doi: 10.1080/00207543.2014.910628.

[94] M. M. Prasad and K. P. Rao, "Simultaneous Scheduling of Machines, Tools and Jobs in Multimachine Flexible Manufacturing System by Using Black Widow Optimization Algorithm," *Studies in Computational Intelligence*, vol. 1027, pp. 423–436, 2022, doi: 10.1007/978-3-030-96634-8_40.

[95] P. Sreenivas, S. khajapeer Saheb, and M. Yohan, "A Trajectory Based Methods for Parallel Scheduling of Machines and AGV'S in an FMS Environment," *IOSR Journal of Mechanical and Civil Engineering*, vol. 14, no. 03, pp. 17–22, 2017, doi: 10.9790/1684-1403021722.

[96] M. Nageswararao, K. Narayanarao, and G. Ranagajanardhana, "Simultaneous Scheduling of Machines and AGVs in Flexible Manufacturing System with Minimization of Tardiness Criterion," *Procedia Materials Science*, vol. 5, no. 4, pp. 1492–1501, 2014, doi: 10.1016/j.mspro.2014.07.336.

[97] P. Singh and A. Jayant, "Selection of Flexible Manufacturing System (FMS): A Synthesis of Three Multi Criteria Decision Making Approaches," *Journal of Material Science and Mechanical Engineering (JMSME)*, vol. 2, no. 8, pp. 15–20, 2015.

[98] S. K. Saren, F. Blaga, S. Dzitac, and T. Vesselenyi, "Decision Based Modeling of a Flexible Manufacturing Cell based on Hierarchical Timed Colored Petri Nets," *Procedia Comput Sci*, vol. 122, pp. 253–260, 2017, doi: 10.1016/j.procs.2017.11.367.

[99] S. Gothwal and T. Raj, "Prioritising the performance measures of FMS using multi-criteria decision making approaches," *International Journal of Process Management and Benchmarking*, vol. 8, no. 1, p. 59, 2018, doi: 10.1504/IJPMB.2018.088657.

[100] V. K. Chawla, A. K. Chanda, S. Angra, and S. Rani, "Simultaneous Dispatching and Scheduling of Multi-Load AGVs in FMS-A Simulation Study," *Mater Today Proc*, vol. 5, no. 11, pp. 25358–25367, 2018, doi: 10.1016/j.matpr.2018.10.339.

[101] R. Erol, C. Sahin, A. Baykasoglu, and V. Kaplanoglu, "A multi-agent based approach to dynamic scheduling of machines and automated guided vehicles in manufacturing systems," *Appl Soft Comput*, vol. 12, no. 6, pp. 1720–1732, Jun. 2012, doi: 10.1016/j.asoc.2012.02.001.

[102] S. Kumar and S. Jain, "Comparison of Dynamic Scheduling Techniques in Flexible Manufacturing System," *Sunil Kumar Int. Journal of Engineering Research and Applications*, vol. 5, no. 7, pp. 143–146, 2015.

[103] S. Burnwal and S. Deb, "Scheduling optimization of flexible manufacturing system using cuckoo search-based approach," *The International Journal of Advanced Manufacturing Technology*, vol. 64, no. 5–8, pp. 951–959, Feb. 2013, doi: 10.1007/s00170-012-4061-z.

[104] B. Huang, R. Jiang, and G. Zhang, "Search strategy for scheduling flexible manufacturing systems simultaneously using admissible heuristic functions and nonadmissible heuristic functions," *Comput Ind Eng*, vol. 71, pp. 21–26, May 2014, doi: 10.1016/j.cie.2014.02.010.

[105] Ö. Başak and Y. E. Albayrak, "Petri net based decision system modeling in real-time scheduling and control of flexible automotive manufacturing systems," *Comput Ind Eng*, vol. 86, pp. 116–126, Aug. 2015, doi: 10.1016/j.cie.2014.09.024.

[106] V. K. Chawla, A. K. Chanda, S. Angra, and S. Rani, "Simultaneous Dispatching and Scheduling of Multi-Load AGVs in FMS-A Simulation Study," *Mater Today Proc*, vol. 5, no. 11, pp. 25358–25367, 2018, doi: 10.1016/j.matpr.2018.10.339.

[107] S. A. V. S. Kumar, V. Veeranna, and B. Durgaprasad, "Multi Objective Scheduling of Jobs in Flexible Manufacturing System Using Metaheuristic Approaches with Inclusion of Simulation Modeling," *International Journal of Computer Engineering in Research Trends*, vol. 351, no. 4, pp. 2349–7084, 2015.

[108] M. Mousavi, H. J. Yap, S. N. Musa, F. Tahriri, and S. Z. Md Dawal, "Multi-objective AGV scheduling in an FMS using a hybrid of genetic algorithm and particle swarm optimization," *PLoS One*, vol. 12, no. 3, p. e0169817, Mar. 2017, doi: 10.1371/journal.pone.0169817.

[109] R. S. Mishra, G. Devi, and A. K. Madan, "International Journal of Research in Engineering and Innovation Multi-objective scheduling in flexible manufacturing systems (FMS) using meta heuristics approach," *International Journal of Research in Engineering and Innovation*, vol. 6, no. 6, pp. 35–41, 2017.

[110] N. Kamatchi and D. Saravanan, "Multi – Objective Flexible Open Shop Scheduling Problem Using Modified Discrete Firefly Algorithm (Mfossp)," *Int J Res Eng Technol*, vol. 5, no. 3, pp. 27–42, 2017.

[111] H. K. Niazi, S. Hou-Fang, Z. Fa-Ping, and R. Ahmed, "Extending MATLAB and GA to solve job shop manufacturing scheduling problems," in *ISPRA'06 Proceedings of the 5th WSEAS International Conference on Signal Processing, Robotics and Automation*, 2006, pp. 345–350.

[112] P. Kumar, P. Kumar, R. Bhool, and V. Upneja, "An efficient Genetic Algorithm Approach for Minimising the Makespan of Job Shop Scheduling Problems An efficient Genetic Algorithm Approach for Minimising the Makespan of Job Shop Scheduling Problems," *International Journal of Science, Engineering and Technology Research*, vol. 5, no. December, pp. 1439–1447, 2017.

[113] O. J. Shukla, G. Soni, A. Sujil, and R. Kumar, "Discrete event system framework for analysis and modeling of job shop scheduling system," in *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Sep. 2016, pp. 2420–2424. doi: 10.1109/ICACCI.2016.7732419.

[114] S. K. Rathore and P. Mishra, "Optimization of Manufacturing Process Using Genetic Algorithm By Optimal Machine Combination : Matlab Simulation & Results," *International Journal of Mechanical Engineering and Technology (IJMET)*, vol. 9, no. 6, pp. 319–333, 2018.

[115]. T Veeraiah, Y. P. Reddy, P. V. S. M. Kumar., and . P W D S Milton, "Optimization of Flow Shop Scheduling by MATLAB," *SSRG International Journal of Mechanical Engineering*, pp. 222–226, 2017.

[116] M. Blaho, M. Foltin, P. Nagy, M. Hudá, and K. Kop, "Operations Scheduling for Manufacturing Systems With Parallel Computing," pp. 3–8, 2009.

[117] Rashmi and S. Bansal, “Task Scheduling of Automated Guided Vehicle in Flexible Manufacturing System using Ant Colony Optimization,” *Int J Latest Trends Eng Technol*, vol. 4, no. 1, pp. 177–181, 2014.

[118] A. Mehrabian, R. Tavakkoli-Moghaddam, and K. Khalili-Damaghani, *Multi-objective routing and scheduling in flexible manufacturing systems under uncertainty*, vol. 14. 2017.

[119] P. Udhayakumar and S. Kumanan, “Sequencing and scheduling of job and tool in a flexible manufacturing system using ant colony optimization algorithm,” *The International Journal of Advanced Manufacturing Technology*, vol. 50, no. 9–12, pp. 1075–1084, Oct. 2010, doi: 10.1007/s00170-010-2583-9.

[120] N. Mathew and N. R. Saravanan, “Evaluation of Genetic Algorithm Approach for Scheduling Optimization of Flexible Manufacturing Systems,” vol. 2, no. 4, pp. 437–446, 2012.

[121] A. V. S. S. Kumar, V. Veeranna, B. Durgaprasad, and B. D. Sarma, “Combined Objective Optimization Of FMS Scheduling With Non-Traditional Optimization Techniques,” *International Journal of Engineering Research & Technology (IJERT*, vol. 2, no. 8, pp. 1420–1428, 2013.

[122] N. M. Nidhiry and R. Saravanan, “Scheduling Optimization of FMS Using NSGA-II,” *Advances in Industrial Engineering and Management*, vol. 3, no. 1, pp. 63–72, 2014, doi: 10.7508/AIEM-V3-N1-63-72.

[123] M. K. Marichelvam, Ö. Tosun, and M. Geetha, “Hybrid monkey search algorithm for flow shop scheduling problem under makespan and total flow time,” *Appl Soft Comput*, vol. 55, pp. 82–92, Jun. 2017, doi: 10.1016/j.asoc.2017.02.003.

[124] I. Fister, X. S. Yang, and J. Brest, “A comprehensive review of firefly algorithms,” *Swarm Evol Comput*, vol. 13, pp. 34–46, 2013, doi: 10.1016/j.swevo.2013.06.001.

[125] S. Mohammadi, B. Mozafari, S. Solimani, and T. Niknam, “An Adaptive Modified Firefly Optimisation Algorithm based on Hong’s Point Estimate Method to optimal operation management in a microgrid with consideration of uncertainties,” *Energy*, vol. 51, pp. 339–348, 2013, doi: 10.1016/j.energy.2012.12.013.

[126] N. Álvarez-Gil, R. Rosillo, D. de la Fuente, and R. Pino, “A discrete firefly algorithm for solving the flexible job-shop scheduling problem in a make-to-order manufacturing system,” *Cent Eur J Oper Res*, 2020, doi: 10.1007/s10100-020-00701-w.

[127] Q. X. Lieu, D. T. T. Do, and J. Lee, “An adaptive hybrid evolutionary firefly algorithm for shape and size optimization of truss structures with frequency constraints,” *Comput Struct*, vol. 195, pp. 99–112, 2018, doi: 10.1016/j.compstruc.2017.06.016.

[128] A. H. Gandomi, X. S. Yang, S. Talatahari, and A. H. Alavi, “Firefly algorithm with chaos,” *Commun Nonlinear Sci Numer Simul*, vol. 18, no. 1, pp. 89–98, 2013, doi: 10.1016/j.cnsns.2012.06.009.

[129] S. Yu, S. Yang, and S. Su, “Self-adaptive step firefly algorithm,” *J Appl Math*, vol. 2013, no. 1, pp. 1–9, 2013, doi: 10.1155/2013/832718.

[130] S. Yu, S. Su, Q. Lu, and L. Huang, “A novel wise step strategy for firefly algorithm,” *Int J Comput Math*, vol. 91, no. 12, pp. 2507–2513, 2014, doi: 10.1080/00207160.2014.907405.

[131] S. Yu, S. Zhu, Y. Ma, and D. Mao, “A variable step size firefly algorithm for numerical optimization,” *Appl Math Comput*, vol. 263, pp. 214–220, 2015, doi: 10.1016/j.amc.2015.04.065.

[132] N. Diab and E. El-Sharkawy, “Recent Advances in Flower Pollination Algorithm,” *International Journal of Computer Applications Technology and Research*, vol. 5, no. 6, pp. 338–346, 2016, doi: 10.7753/ijcatr0506.1003.

[133] Z. A. A. Alyasseri, A. T. Khader, M. A. Al-Betar, M. A. Awadallah, and X. S. Yang, “Variants of the flower pollination algorithm: A review,” *Studies in Computational Intelligence*, vol. 744, no. January, pp. 91–118, 2018, doi: 10.1007/978-3-319-67669-2_5.

[134] W. Cui and Y. He, “Biological Flower Pollination Algorithm with Orthogonal Learning Strategy and Catfish Effect Mechanism for Global Optimization Problems,” *Math Probl Eng*, vol. 2018, 2018, doi: 10.1155/2018/6906295.

[135] C. Qu, Y. Fu, Z. Yi, and J. Tan, “Solutions to no-wait flow shop scheduling problem using the flower pollination algorithm based on the hormone modulation mechanism,” *Complexity*, vol. 2018, 2018, doi: 10.1155/2018/1973604.

[136] K. Gayathri Devi, R. S. Mishra, and A. K. Madan, “Combined Objective Optimization of FMS Scheduling by a Hybrid Genetic Algorithm,” in *Advances in Mechanical and Materials Technology*, 2022, pp. 125–139. doi: https://doi.org/10.1007/978-981-16-2794-1_11.

[137] K. Gayathri Devi, R. S. Mishra, and A. K. Madan, “A Dynamic Adaptive Firefly Algorithm for Flexible Job Shop Scheduling,” *Intelligent Automation & Soft Computing*, vol. 31, no. 1, pp. 429–448, 2022, doi: 10.32604/iasc.2022.019330.

[138] G. D. Karunagaran, R. S. Mishra, A. K. Madan, and A. Engineering, “INTEGRATED APPROACH OF SCHEDULING A FLEXIBLE JOB SHOP USING ENHANCED FIREFLY AND HYBRID FLOWER POLLINATION,” *International Journal of Industrial Engineering: Theory, Application and Practice*, vol. 29, no. 6, pp. 772–792, 2022, doi: 10.23055/ijietap.2022.29.6.8291.

[139] X. Yang, *Nature-Inspired Metaheuristic Algorithms Second Edition*, no. May. 2010. doi: 10.1016/j.chemosphere.2009.08.026.

[140] W. Xu, Z. Ji, and Y. Wang, “A flower pollination algorithm for flexible job shop scheduling with fuzzy processing time,” *Modern Physics Letters B*, vol. 32, no. 34–36, 2018, doi: 10.1142/S0217984918401139.

[141] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms: Second Edition*. Luniver Press, 2010.

[142] N. J. Cheung, X. M. Ding, and H. bin Shen, “Adaptive firefly algorithm: Parameter analysis and its application,” *PLoS One*, vol. 9, no. 11, 2014, doi: 10.1371/journal.pone.0112634.

[143] M. Sababha, M. Zohdy, and M. Kafafy, “The enhanced firefly algorithm based on modified exploitation and exploration mechanism,” *Electronics (Switzerland)*, vol. 7, no. 8, 2018, doi: 10.3390/electronics7080132.

[144] M. R. Singh and S. S. Mahapatra, “A quantum behaved particle swarm optimization for flexible job shop scheduling,” *Comput Ind Eng*, vol. 93, pp. 36–44, 2016, doi: 10.1016/j.cie.2015.12.004.

[145] I. Kacem, S. Hammadi, and P. Borne, “Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems,” *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 32, no. 1, pp. 1–13, 2002, doi: 10.1109/TSMCC.2002.1009117.

[146] S. Dauzère-Pérès and J. Paulli, “An integrated approach for modeling and solving the general multiprocessor job-shop scheduling problem using tabu search,” *Ann Oper Res*, vol. 70, pp. 281–306, 1997, doi: 10.1023/A:1018930406487.

[147] P. Brandimarte, “Routing and scheduling in a flexible job shop by tabu search,” *Ann Oper Res*, vol. 41, no. 3, pp. 157–183, 1993, doi: 10.1007/BF02023073.

[148] X. Du, Z. Li, and W. Xiong, “Flexible Job Shop scheduling problem solving based on genetic algorithm with model constraints,” *2008 IEEE International Conference on Industrial Engineering and Engineering Management, IEEM 2008*, pp. 1239–1243, 2008, doi: 10.1109/IEEM.2008.4738068.

[149] M. Rajkumar, P. Asokan, N. Anilkumar, and T. Page, “A GRASP algorithm for flexible job-shop scheduling problem with limited resource constraints,” *Int J Prod Res*, vol. 49, no. 8, pp. 2409–2423, 2011, doi: 10.1080/00207541003709544.

[150] G. Balaraju, S. Venkatesh, and B. S. P. Reddy, “Multi-objective flexible job shop scheduling using hybrid differential evolution algorithm,” *International Journal of Internet Manufacturing and Services*, vol. 3, no. 3, pp. 226–243, 2014, doi: 10.1504/IJIMS.2014.062436.

[151] S. Huang, N. Tian, and Z. Ji, “Particle swarm optimization with variable neighborhood search for multiobjective flexible job shop scheduling problem,” *International Journal of Modeling, Simulation, and Scientific Computing*, vol. 07, no. 03, p. 1650024, 2016, doi: 10.1142/S1793962316500240.

[152] Q. Deng, G. Gong, X. Gong, L. Zhang, W. Liu, and Q. Ren, “A Bee Evolutionary Guiding Nondominated Sorting Genetic Algorithm II for Multiobjective Flexible Job-Shop Scheduling,” *Comput Intell Neurosci*, vol. 2017, 2017, doi: 10.1155/2017/5232518.

[153] E. R. R. Kato, G. D. de A. Aranha, and R. H. Tsunaki, “A new approach to solve the flexible job shop problem based on a hybrid particle swarm optimization and Random-Restart Hill Climbing,” *Comput Ind Eng*, vol. 125, pp. 178–189, 2018, doi: 10.1016/j.cie.2018.08.022.

[154] T. hua Jiang and C. Zhang, “Adaptive discrete cat swarm optimisation algorithm for the flexible job shop problem,” *International Journal of Bio-Inspired Computation*, vol. 13, no. 3, pp. 199–208, 2019, doi: 10.1504/ijbic.2019.10020503.

[155] Y. Zhang, H. Zhu, and D. Tang, “An improved hybrid particle swarm optimization for multi-objective flexible job-shop scheduling problem,” *Kybernetes*, no. 3082018, 2019, doi: 10.1108/K-06-2019-0430.

[156] J. Gao, L. Sun, and M. Gen, “A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems,” *Comput Oper Res*, vol. 35, no. 9, pp. 2892–2907, 2008, doi: 10.1016/j.cor.2007.01.001.

[157] W. T. Lunardi, L. H. Cherri, and H. Voos, “A Mathematical Model and a Firefly Algorithm for an Extended Flexible Job Shop Problem with Availability Constraints,” *Springer International Publishing AG, part of Springer Nature 2018*, vol. 1, no. 2015, pp. 548–560, 2018, doi: 10.1007/978-3-319-91253-0.

[158] L.-N. Xing, Y.-W. Chen, and K.-W. Yang, “An efficient search method for multi-objective flexible job shop scheduling problems,” *J Intell Manuf*, vol. 20, no. 3, pp. 283–293, 2009, doi: 10.1007/s10845-008-0216-z.

[159] J. Q. Li, Q. K. Pan, and Y. C. Liang, “An effective hybrid tabu search algorithm for multi-objective flexible job-shop scheduling problems,” *Comput Ind Eng*, vol. 59, no. 4, pp. 647–662, 2010, doi: 10.1016/j.cie.2010.07.014.

[160] S. Karthikeyan, P. Asokan, and S. Nickolas, “A hybrid discrete firefly algorithm for multi-objective flexible job shop scheduling problem with limited resource constraints,” *International Journal of Advanced Manufacturing Technology*, vol. 72, no. 9–12, pp. 1567–1579, Jun. 2014, doi: 10.1007/s00170-014-5753-3.

[161] J. Jerald, P. Asokan, G. Prabaharan, and R. Saravanan, “Scheduling optimisation of flexible manufacturing systems using particle swarm optimisation algorithm,” *International Journal of Advanced Manufacturing Technology*, vol. 25, no. 9–10, pp. 964–971, 2005, doi: 10.1007/s00170-003-1933-2.

[162] Dr. R. S. Nidhish Mathew Nidhiry, “A COMBINED OBJECTIVE GENETIC ALGORITHM FOR SCHEDULING OF MACHINES IN FMS,” *International Journal of Engineering Research and Applications (IJERA) ISSN:*, vol. 3, no. 2, 2012.

[163] N. M. Nidhiry and R. Saravanan, “Scheduling Optimization of FMS Using NSGA-II,” *Advances in Industrial Engineering andManagement*, vol. 3, no. 1, pp. 63–72, 2014, doi: 10.7508/AIEM-V3-N1-63-72.

[164] Ü. Bilge and G. Ulusoy, “A TIME WINDOW APPROACH TO SIMULTANEOUS SCHEDULING OF MACHINES AND MATERIAL HANDLING SYSTEM IN AN FMS,” *Oper Res*, vol. 43, no. 6, pp. 1058–1070, 1995, doi: 10.1017/CBO9781107415324.004.

[165] G. Ulusoy, F. Sivrikaya-Şerifoğlu, and Ü. Bilge, “A genetic algorithm approach to the simultaneous scheduling of machines and automated guided vehicles,” *Comput Oper Res*, vol. 24, no. 4, pp. 335–351, 1997, doi: 10.1016/S0305-0548(96)00061-5.

[166] T. F. Abdelmaguid, A. O. Nassef, B. A. Kamal, and M. F. Hassan, “A hybrid GA/heuristic approach to the simultaneous scheduling of machines and automated guided vehicles,” *Int J Prod Res*, vol. 42, no. 2, pp. 267–281, 2004, doi: 10.1080/0020754032000123579.

[167] B. S. P. Reddy and C. S. P. Rao, “A hybrid multi-objective GA for simultaneous scheduling of machines and AGVs in FMS,” *International Journal of Advanced Manufacturing Technology*, vol. 31, no. 5–6, pp. 602–613, 2006, doi: 10.1007/s00170-005-0223-6.

[168] L. Deroussi, M. Gourgand, and N. Tchernev, “A simple metaheuristic approach to the simultaneous scheduling of machines and automated guided vehicles,” *Int J Prod Res*, vol. 46, no. 8, pp. 2143–2164, 2008, doi: 10.1080/00207540600818286.

[169] Dr. R. D. and Dr. P. R. K. Sivarami Reddy N, “Simultaneous Scheduling of Machines and Agvs Using Flower Pollination Algorithm: a New Nature-Inspired Meta-Heuristic,” *Manufacturing Technology Today*, vol. 17, no. 7, pp. 19–30, 2018.

LIST OF PUBLICATIONS

SCI Journals:

1. K. Gayathri Devi, R. S. Mishra and A. K. Madan, "A dynamic adaptive firefly algorithm for flexible job shop scheduling," *Intelligent Automation & Soft Computing*, vol. 31, no.1, pp. 429–448, 2022. https://doi.org/10.32604/iasc.2022.019330
2. K. Gayathri Devi, R. S. Mishra and A. K. Madan, "Integrated Approach of Scheduling a Flexible Job Shop using Enhanced Firefly and Hybrid Flower Pollination Algorithms", *International Journal of Industrial Engineering: Theory, Applications and Practice*, *29*(6). https://doi.org/10.23055/ijietap.2022.29.6.8291

ESCI Journal:

1. K. Gayathri Devi, R. S. Mishra and A. K. Madan, "A Dynamic Hybrid Firefly Algorithm to Minimize Make-span for Flexible Job Shop Scheduling" (in press) **Advances in Materials and Processing Technologies (AMPT), Taylor & Francis**

SCOPUS Journals:

1. Gayathri Devi, K., Mishra, R.S., Madan, A.K. (2022). Combined Objective Optimization of FMS Scheduling by a Hybrid Genetic Algorithm. In: Govindan, K., Kumar, H., Yadav, S. (eds) Advances in Mechanical and Materials Technology . Lecture Notes in Mechanical Engineering. Springer, Singapore. https://doi.org/10.1007/978-981-16-2794-1_11

Non Scopus Journals:

1. Gayathri Devi K, R. S. Mishra, A. K. Madan *"Flower Pollination Algorithm for Scheduling Optimization of Flexible Manufacturing System"* , International Journal of Engineering and Techniques - Volume 8 Issue 3, June 2022 DOI: 10.5281/zenodo.6630708

2. Gayathri Devi K, R. S. Mishra, A. K. Madan.*"A Hybrid Firefly Algorithm Approach for Job Shop Scheduling Problem "*, Volume 9, Issue XII, International Journal for Research in Applied Science and Engineering Technology (IJRASET) Page No: 1436-1444, ISSN : 2321-9653, https://doi.org/10.22214/ijraset.2021.39536

3. Gayathri Devi K, R.S. Mishra, A.K Madan, "A Hybrid Firefly Algorithm for Multi Objective Optimization in Flexible Manufacturing System", *International Research Journal of Engineering and Technology (IRJET)* Vol 7, Issue 5 (2020), 822-829.
4. Gayathri Devi, R.S. Mishra, A.K Madan, "Non-traditional optimization techniques of scheduling in FMS: A Review", *International Journal of Research in Engineering and Innovation* Vol-3, Issue-2 (2019), 157-164
5. Mishra, RS., & Gayathri Devi, et al. (2017). "Multi-objective scheduling in flexible manufacturing systems (FMS) using meta heuristics approach" . *International Journal of Research in Engineering and Innovation,1*(6), 35-41.

CONFERENCES

1. "A dynamic Hybrid Firefly Algorithm to minimize make-span for Flexible Job shop Scheduling" in International Conference on Advance Production and industrial Engineering (ICAPIE-2021), Delhi.
2. "Combined Objective Optimization of FMS Scheduling by a hybrid Genetic Algorithm" in International Conference on Energy, Materials Sciences and Mechanical Engineering (EMSME-2020), Delhi (2020)
3. "Scheduling a Job Shop FMS by a Hybrid Metaheuristic Approach" in International Conference of Advance Mechanical Engineering Research and Innovation (ICARI-2020)
4. "Optimisation of Scheduling a Job Shop FMS by a Hybrid Technique" in 25th International Multi-Disciplinary Conference on Science, Management and Technology, Malaysia (2019)

Under Review

1. "An Improved Flower Pollination Algorithm for Energy Cognizant Flexible Job Shop Scheduling Problem" submitted in Facta Universitatis, Series: Mechanical Engineering.