OPTIMIZATION OF SCHEDULING TECHNIQUES IN FMS IN THE CONTEXT OF INDIAN INDUSTRY

A THESIS

SUBMITTED IN FULFILLMENT OF THE

REQUIREMENTS FOR THE AWARD OF THE DEGREE

OF

DOCTOR OF PHILOSOPHY

IN

MECHANICAL ENGINEERING

By

GAGANPREET KAUR

(2K16/Ph.D/ME/43)

UNDER THE SUPERVISION OF

Prof. R. S. MISHRA

Prof.A.K.MADAN

DEPARTMENT OF MECHANICAL ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-110042, India

February, 2026

CANDIDATE'S DECLARATION

I hereby declare that the research work presented in this thesis, entitled **"Optimization Of Scheduling Techniques in FMS in the Context of Indian Industry"** is an original work carried out by me under the supervision of Prof. R.S. Mishra., Professor and Prof.A.K.Madan 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:

Gaganpreet Kaur
(2K16/Ph.D/ME/43)
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, **"Optimization Of Scheduling Techniques in FMS in the Context of Indian Industry "** being submitted by **Gaganpreet Kaue (Roll No- 2K16/PhD/ME/43)** 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)

Prof.A.K.Madan
CO-SUPERVISOR
Professor, Dept. of Mechanical Engineering (DTU, Delhi)

ACKNOWLEDGEMENT

This thesis was completed during my tenure as a doctoral candidate at Delhi Technological University, Delhi. First and foremost, I express my heartfelt gratitude to the Almighty, whose boundless mercy and blessings granted me the strength and perseverance to embark on this journey of knowledge and discovery.

I am deeply grateful to my supervisor, Prof. R.S. Mishra, for his steadfast support and invaluable guidance throughout my Ph.D. journey. His patience, motivation, deep insights, and enthusiasm were instrumental in shaping my research. I feel truly privileged to have received his mentorship, which greatly enriched both the research process and the writing of this thesis. His academic feedback and thoughtful perspectives on research and scholarly writing have been particularly impactful.

I would also like to sincerely thank my Co-Supervisor, Prof. A.K. Madan, for his constant encouragement and for sharing his rich expertise. His thoughtful input was critical during the more challenging phases of the research, and his guidance made my work more refined and purposeful.

My sincere thanks go to the esteemed members of the dissertation committee. I am immensely thankful for their time, insightful feedback, and constructive questions, all of which contributed meaningfully to the refinement of this research. I also extend my gratitude to the Department of Mechanical Engineering at Delhi Technological University, Delhi, for providing the infrastructure and support necessary to carry out this work.

Research at the doctoral level is never a solitary endeavor. I am especially thankful to my colleagues, Mrs. Gayatri Devi and Dr. Manoj Yadav, for their constant moral support and for patiently listening to my frequent discussions about research.

I owe everything to my incredible parents, Sardar Harbans Singh and Ravinder Kaur, whose unwavering faith in me, boundless encouragement, and sacrifices gave me the strength to pursue my dreams. Their unconditional love has always been a source of courage.

To my husband, Ravi—thank you from the bottom of my heart for standing by me through every phase of this journey. Your unshakeable support, belief in my abilities, and quiet strength have meant everything to me. You have truly been my anchor.

I'm also grateful to my brother, Deepinder, who has always been by my side during difficult times, often shouldering my family responsibilities so I could focus on my academic path with peace of mind.

Lastly, to my dear child, Anahita—you are my greatest inspiration. Your love, laughter, and

joyful spirit gave me energy and hope during the most demanding periods. Even when I couldn't be fully present, your understanding and support never wavered. I hope this achievement serves as an example to you to follow your dreams with determination and passion. Words will never be enough to express how much I love you.

ABSTRACT

In today's rapid and highly demanding production environment, manufacturing enterprises are increasingly adopting Flexible Manufacturing Systems (FMS) to enhance productivity, responsiveness, and operational efficiency. An FMS represents an integrated and automated manufacturing environment that combines programmable machine tools, automated material handling systems, automated storage and retrieval units, and centralized computer control to efficiently manufacture a wide variety of components. As a cornerstone of Industry 4.0, FMS plays a critical role in enabling intelligent, data-driven, and adaptive manufacturing systems capable of responding to dynamic market demands.

A fundamental challenge in FMS operation is scheduling optimization, which involves determining the optimal sequence of jobs while allocating limited and interdependent resources such as machines, tools, and Automated Guided Vehicles (AGVs). Effective scheduling significantly enhances system performance by reducing make span, minimizing tardiness, improving resource utilization, lowering operational costs, and increasing throughput. In the context of Industry 4.0, scheduling optimization becomes even more crucial, as it directly supports the realization of smart, autonomous, and energy-efficient manufacturing environments. Traditional optimization approaches such as linear programming, dynamic programming, and exact mathematical models have been widely used for manufacturing scheduling. However, due to the combinatorial complexity, non-linearity, multi-objective nature, and large search spaces inherent in FMS, these methods often become computationally infeasible for real-world applications. In contrast, metaheuristic algorithms, inspired by natural and evolutionary processes, provide robust and scalable alternatives. Their ability to balance exploration and exploitation, handle multiple conflicting objectives, and adapt to complex constraints makes them particularly suitable for FMS scheduling problems. With recent advances in computational power and intelligent optimization strategies, metaheuristics have emerged as powerful tools for addressing large-scale industrial scheduling challenges.

In this research, three novel hybrid metaheuristic methodologies are proposed to address simultaneous scheduling problems in FMS under Industry 4.0 paradigms:

1. Dynamic-Particle Multi-Swarm Optimization (Dy-PSO)
2. Novel Variant of Particle Swarm Optimization (NvPSO) and Walrus Optimization Algorithm (WaOA)
3. Novel Genetic and Adaptive Artificial Bee Colony Algorithm (NG-AABCA)

The first study introduces Dynamic-Particle Multi-Swarm Optimization (Dy-PSO), a novel scheduling framework primarily focused on makespan minimization in FMS. Dy-PSO employs multiple interacting swarms with adaptive parameter control to prevent premature convergence and stagnation. A spatial exclusion strategy is incorporated to avoid redundant exploration of previously visited regions of the solution space. A significant methodological contribution of this study is the integration of machine learning, where a Random Forest Regressor, enhanced through Genetic Algorithm–based learning, is used to predict and guide scheduling decisions. This hybridization of evolutionary optimization and predictive learning establishes a data-driven scheduling framework that enhances solution quality and convergence speed. Dy-PSO is evaluated under three realistic scheduling scenarios: (i) simultaneous scheduling of jobs, tools, and AGVs, (ii) scheduling without tool constraints, and (iii) scheduling integrated with machine learning assistance. Comparative analysis with conventional PSO demonstrates substantial reductions in makespan and superior robustness across all scenarios.

The second study focuses on energy-aware scheduling through a Novel Variant of PSO (NvPSO) and the Walrus Optimization Algorithm (WaOA). This study addresses the growing need for sustainable manufacturing by simultaneously minimizing makespan, tardiness, and total energy consumption. NvPSO incorporates logistic map–based parameter tuning, enhancing diversity and search efficiency. Experimental results across 13 diverse job sets show that NvPSO achieves up to 9% reduction in energy consumption while completely eliminating tardiness penalties, outperforming conventional algorithms such as the Artificial Immune System (AIS) and Modified Genetic Tabu Algorithm (MGTA). While WaOA demonstrates faster convergence and lower computational complexity, NvPSO consistently delivers superior solution quality for larger and more complex scheduling instances, highlighting its suitability for energy-efficient Industry 4.0 manufacturing systems.

The third study proposes the Novel Genetic and Adaptive Artificial Bee Colony Algorithm (NG-AABCA) for minimizing makespan, penalty costs, and total tardiness. NG-AABCA introduces cognitive (ε_1) and social (ε_2) learning mechanisms, which are generally underutilized in classical ABC algorithms, to exploit global knowledge and enhance convergence behavior. The algorithm further integrates Genetic Algorithm elitism and Random-Restart Hill-Climbing, effectively balancing solution diversity and intensification. Computational results demonstrate that NG-AABCA achieves a 5.3% reduction in makespan and an 8.7% reduction in tardiness compared to conventional metaheuristics, resulting in improved productivity and more efficient utilization of manufacturing resources.

Extensive computational experiments were conducted using MATLAB R2019a on an Intel Core™ i7 platform, and the proposed algorithms were validated across benchmark datasets as well as realistic FMS configurations. The results confirm that the suggested hybrid metaheuristic approaches consistently outperform traditional optimization methods in terms of solution quality, convergence speed, robustness, and scalability. In several cases, the algorithms identified new best-known makespan values, demonstrating their effectiveness in exploring high-quality solution spaces.

Overall, this research makes significant contributions by developing adaptive, hybrid, and energy-aware scheduling frameworks that address the complexities of simultaneous resource scheduling in FMS. The integration of metaheuristics, machine learning, and sustainability objectives positions the proposed approaches as powerful decision-support tools for Industry 4.0–enabled smart manufacturing, offering both theoretical advancements and strong potential for industrial applicability.

.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

IST 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
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
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
PSM	Particle Swarm Method
SA	Simulated Annealing
DR	Dispatching Rules
FJSSP	Flexible job shop scheduling problem
COF	Combined Objective Function
DE	Differential Evolution
MOPSO	Multi-Objective particle swarm optimization
RRHC	Random Restart Hill Climbing
DP DATA	Dauzère–Pérès data set

CHAPTER 1

INTRODUCTION

In today's competitive global market, businesses must modify their operations to ensure that they can respond to client requirements more quickly and effectively. The primary goal of any manufacturing firm is to reach a high level of productivity and versatility, which can only be accomplished with the aid of computer integrated manufacturing. A flexible manufacturing system (FMS) provides a small degree of flexibility, allowing the device to adjust to changes, whether planned or not. FMS comprises three essential arrangements. A material handling system (MHS) connects the CNC machines to optimise part flow, while the central control computer gearshifts material motions and machine flow.

1.1. Flexible manufacturing system

It is a system with several programmable machine tools connected by an automated material handling system that can generate a wide range of products. An FMS is enormous, complicated, and expensive, with computers controlling all of the devices. This is why many firms prefer smaller versions known as flexible manufacturing cells over FMS. Nowadays, two or more CNC machines are designated a Flexible Manufacturing Cell (FMC), while two or more cells are considered a Flexible Manufacturing System (FMS).

“Flexible manufacturing system is a computer-controlled manufacturing system, in which numerically controlled machines are interconnected by a material handling system and a master computer controls both NC machines and material handling system.” [1]

The primary goal of a manufacturing industry is to improve efficiency through increased throughput, flexibility, and system utilisation. System utilization is calculated as a percentage of the available hours (Number of the machines available for production multiplied by the number of working hours), it can be increased by shifting plant layout or by decreasing FMS transfer time between two stations and throughput is defined as the number of parts produced by the last machine of a manufacturing system over a given period of time. If the number of parts increases throughput also increases and also system utilization increases. Flexible manufacturing system is divided into components:

Work station: It consists of CNC machines that perform various operations on group of parts.

Automated Material Handling and Storage System: Work parts and subassembly parts between the processing stations are transported by several automated MHS. Many automated material handling devices, such as automated guided vehicles and conveyors, are used in flexible manufacturing systems.

The computer control system regulates the actions of the FMS's processing stations and material handling system.

Job sequences

In a specific FMS environment, all machines are organised in a distinct manner. The collection of jobs is processed with specific operations. Keeping in mind their processing duration, these jobs' due dates have been set to minimise make span. To produce the ideal sequence, the rules are chosen from a large number of current priority scheduling rules.

First-Come, First-Serve (FCFS) - The activity that arrives first enters service first (local regulation). It is straightforward, fast, and "honest" with the customer. The disadvantage of this rule is that it is the least effective when measured using standard performance measures because a protracted job causes others to wait, resulting in idle downstream resources, and it ignores job due date and work remaining.

Shortest Processing Time (SPT) - The task with the shortest operation time enters carrier first. In general, it is a superior rule utilised in minimising completion time, minimising the average number of jobs in the system, especially reducing in-system inventories (reduced shop congestion) and downstream idle time (greater aid consumption). Typically, task tardiness is reduced, and negatives include ignorance of downstream, due date information, and lengthy job wait times.

Earliest Due Date (EDD) - The task with the closest due date enters first (nearby rule), and it is a fast and simple rule that typically performs well in terms of due date, but there is no consideration of job processtime.

The Critical Ratio (CR) Rule states that jobs should be sequenced based on the remaining time from the due date divided by the remaining processing time. The job with the smallest CR goes into service first. The ratio is calculated as (Due Date - Present Time) divided by remaining shop time, which includes queue, set-up, run, wait, and move times at present and downstream work centres. It acknowledges the task due date and the amount of work remaining (including downstream data); nonetheless, in this sequencing, past-due jobs are given top priority, regardless of the number of outstanding operations.

Slack Per Operation - is a global rule, where job priority is determined as (Slack of remaining operations) it recognizes job due date and work remaining (incorporates downstream information)

Least Changeover Cost (Next Best rule) -It sequences jobs by set-up cost or time (local rule).

It is simple, fast, and generally performs well with regards to set-up costs. it does not consider the job process time, due date and work remaining.

1.2 Scheduling

Scheduling problems can be considered appropriately only when questions like - which products and in

what quantities, how to manufacture and what is the availability of resources; have been addressed adequately. Scheduling deals with the allocation of limited resources. The purpose of scheduling is to optimise one or more resources. Scheduling is a decision-making problem which is used by production and service industries. It is also utilized in procurement and manufacturing, as well as transportation and distribution. Organisational scheduling use mathematical or heuristic procedures to allocate limited resources to task processing. It can be used for multiple resources and might have extraordinary priorities. Manufacturers are committed to maintain imparting completed products to purchaser on due dates. There can be differing situations with diverse results and it can create demand pressure on available resources. So timely choice becomes crucial during everyday operations. As a result, scheduling is a decision-making function: it is the ongoing process of defining a schedule. Scheduling is a corpus of theory, consisting of principles, models, procedures, and logical conclusions that provide insight into the scheduling function.

Scheduling can be defined as, “Establishing the timing of performing each task in a set”. Cox et al., (1992) define detailed scheduling as "the actual assignment of starting and/or completion dates to operations or groups of operations to show when these must be done if the manufacturing order is to be completed on time”. It is also called as Operations Scheduling, (Herrman, 2006).

1.2.1 Need of Scheduling

As we work in real time, a lot of additional information is available, using which we can make the system more robust and represent reality. The prevalence of arbitrary events like breakdown of a machine or absenteeism of skilled labour, or delays in raw material supplies, is inevitable in a manufacturing system. Issues can be more accurately tackled in short-term planning than medium-term planning. Operations scheduling largely impacts materials requirements planning (MRP) systems. The output of an MRP exercise becomes basis for scheduling of jobs.

Scheduling is described in many ways, conveying the important concepts related to it. The following description is given by Mahdavinejad (2007) a) Loading: assignment of jobs to machines, b) Scheduling: ranking of jobs of each work center, c) Routing: the order in which resources are available for processing, are used by jobs d) Sequencing: ordering of operations of the jobs in production e) Dispatching: As per the sequence or the schedule, authorizing the processing of the jobs by the work center f) Expediting: guarantees the routing and schedule of job travels through various phases g) List scheduling: As per availability of machine listing the operations in some order and then assigning them to the machines (Kim, 1990).

The shop floor is not the only aspect of the business that influences the scheduling process. The scheduling method also interacts with the production planning process, which manages medium to long-

term planning for the entire organisation. This approach aims to optimise the firm's average product mix and long-term useful resource allocation using stock ranges, demand projections, and useful resource requirements. Decisions made at this stage aid in the development of better plans and may have an immediate impact on the more comprehensive scheduling process.

Classification of Scheduling Problems

Scheduling can be categorised into two groups:

1) scheduling used in service organization.

2) scheduling in manufacturing.

Scheduling used in a service organization involves scheduling of resources such healthcare facilities, automobile servicing, maintenance and repairs of machine tools and time-tabling. The techniques used for such type of problems are dissimilar than those for the manufacturing type. Legal considerations such as remunerations, working hours and conditions can constrain scheduling judgments in service organization.

There are two types of manufacturing organizations: continuous process and discrete process. The continuous process type is mainly found in process industry. The bulk of raw materials are processed and scheduling is done in accordance to availability of raw materials, demand, the nature of the process ,and the capacity of the plant. The discrete manufacturing comprises processing at different work centers of individual components. In accordance to a process sequence the workpieces are transferred from one work station to another, as soon as it become free. The automobile manufacturing industry and engineering industries use discrete scheduling. This research is confined to scheduling in discrete parts manufacturing systems.

Four types of information are considered while describing a specific scheduling problem, in discrete manufacturing systems (Conway et al. (2003)

(i) The jobs and the operations to be performed, (ii) The number and types of machines available in the machine shop (iii) Disciplines that constrain how jobs can be assigned to machines (iv) The criteria for evaluation of the schedule.

Batch or mass production depends on the number of jobs taken for processing. All the jobs which arrive simultaneously in a shop is a static problem. All the machines are available to these jobs at start and during this time no further jobs arrive. But in a dynamic problem jobs arrive in accordance to a probability distribution. According to the job arrivals machine availability keeps on varying. Entirely different methods are used to analyze these two types of problems. Static models are easier than dynamic models as they are more challenging than static models.

In flow shop problem, same order of operations of all the jobs are essentially followed. With job variability, the processing times may differ. But in case of a job shop it is not obligatory to have identical operation sequences. Mostly the real shops fall someplace between these two extremes. A third type of problem is open shop problem, wherein the manner in which operations are carried out on the jobs is immaterial (Leung, 2004). A flexible flow shop has the capability to modify the paths of jobs if necessary, in a real time manner, without affecting their operation sequences with the help of programmable machines known as parallel machines (Pinedo (2005)). In addition, it is also possible to change operation sequence and the contents of the operations by merging or dividing the partial operations in real-time. In FMS the variable route part programming system is used which makes it possible and it is used as a basis for the dynamic scheduling system (Ranky, 1984). The manufacturing system though, needs to be fully integrated.

1.2.2 Complex Job Shops

In the job arrival process 4 parametric notation (A/B/C/D) is normally used (Conway et al., 2003) to categorize scheduling problems.

A- For static problems it is a fixed but random number of jobs (n) that can arrive instantaneously. For dynamic arrivals, it is identified by the probability distribution of time between job arrivals.

B- The no of machines (m) in the shop.

C- The flow arrangement of the jobs in the shop. F for flow shops, G for random patterns and R for arbitrarily routed job shops. It is not suited to single-machine problems.

D- The criteria for estimation of the schedule. F for flow time, N for number of jobs, L for lateness, C- for makespan, and U for utilization of machines in system at a time,t.

Measures of Schedule Evaluation

The commonly used criteria for schedule estimation are:

- **i)** Flow time: time for which a job is on the shop floor, good flow type jobs.
- **ii)** Makespan: It is the completion time of the last job to leave the system.
- **iii)** Machine utilization: the amount of time during which the machine is busy processing jobs .
- **iv)** Lateness / Earliness: difference between the completion time and the due date for a job.
- **v)** Number of tardy jobs: focuses on whether a job is tardy (late) at all, and not on the amount of lateness. The performance of managers is accounted with help of no of tardy jobs.
- **vi)** Setup time/cost: the time needed for setting up job and cost of raw material wasted during the setup process

vii) Work-in-process (WIP) inventory costs: Cost due to incomplete jobs on the shop floor for large time, which is reflected in the usual number of jobs in the system and average completion time of the jobs at a given time. The monetory computation is ver difficult of the stated measures.

1.2.3 Schedules for Flow Shop Problems

Flow shop difficulties such as two machines and n jobs can be solved using Johnson's rule. Using this technique, the three machines n jobs problem may also be solved optimally. The two jobs m machines problem can be solved graphically. Nevertheless the big problems like n jobs problem can be solved by permutation scheduling problem. Like for a 12 jobs problem there are 12! (479001600) alternative sequences it is quite a big number. So, simulation and metaheuristic procedures can be efficiently used for such problems.

Forward and backward scheduling for flow shops

There are two types of classification of scheduling techniques. Forward scheduling starts as the order is known and the objective is “ASAP’ delivery. A schedule that can be accomplished is made even if the due dates are not met. Scheduling final operation first in reverse order id done in Backward scheduling considering the due dates. The main objective is to meet the due dates. When there is a bottleneck machine combination of both may be used. The bottleneck machine is a single machine scheduling problem.

1.3 Scheduling Context

The nature and the intricacy of the scheduling problem is profoundly dependent on its context. Following parameters describe the scheduling problem effectively (Mahdavinejad, 2007).

- Number of machines (m): In case of a single machine problem (m = 1), the jobs need to be truly rank ordered on some basis. As m turns into huge the problem size will increase as several combos are introduced. If a couple of machines are available for the jobs as parallel sources, we want to expand a way via which one out of these is selected for task.
- Number of jobs (n): Wide no of jobs makes the problem more intricate, since multiple jobs strive for the similar resources and time slots.
- Shop configuration: It is the manner wherein the m machines are organized on the shop floor and the processing sequence is to be followed by the jobs. Flow shop, job shop and open shop configurations

are described in advance.

Open and Flow shop are the handiest configurations among those. (a) In a flow shop the machines are arranged in line with the processing sequence of the jobs, for instance, a transfer line. In case of a mixed flow shop the scheduling complexity increases. (b) Similar types of resources may be grouped collectively in a job shop. Every job follows a completely unique operation sequence to visit the machines. The set of jobs processed at each machine can be exclusive; hence ordering of jobs in front of every machine needs to be determined one by one.

1.3.1 Constraints and Assumptions

The type of manufacturing decides the constraints in a scheduling problem. The constraints of a job shop scheduling are:

1) The preceding operation must be over on a machine on a job before starting an operation. There should be no violation of the processing sequence.

2) The next operation cannot be started on that machine without the present operation on a machine is not completed and the machine becomes “free”.

3) Disruption in an operation is not allowed. In emergencies, if operation is interrupted it results in wasting the interrupted part. The schedule needs to be revised in such cases.

A number of assumptions are made in order to simplify the scheduling problem. One or more of these assumptions may not hold good in real life situations, as indicated.

- Before the manufacturing process starts all the raw materials should be available on the shop floor for continuous availability to machines assigned.
- Jobs should comply with precedence constrain. (exceptions are for FMS, where variable route part programming is possible)
- Only one machine in the shop can perform each operation. This rule is not applicable in case of parallel machines or flexible job shops.
- There is only one machine of each type in the shop. (Not applicable in case of parallel machines or flexible job shops)
- There should be no overlapping of process times of successive operations of a particular job .(In case of flow shop or batch production this rule may not be adhered)
- Processing time, assumed to be deterministic.
- The processing times should include the setup times. (Its amount to the total processing time varies for batch type production with the batch size. Separate considerations are required for

sequence-dependent setup times.

- Each job visits each machine only once. (This assumption is not applicable for versatile machines e.g. CNC machining centers)
- Due dates of the jobs are undefined.
- No interruption until each operation of each job is finished may prevail.

4) There are interdependencies between different jobs, job cannot start until another is completed.

5) Release times and due dates are specified (temporal constraints).

6) Machines can perform several operations with differing efficiencies.

7) Due date penalties can be specified.

8) Multiple scheduling criteria can be optimized.

1.4 Multiplicity of Objectives

There are many objectives for the scheduling problem like "Meeting due dates" and Minimising Makespan" are mainly aimed. Plant efficiency can be achieved by objectives like "Minimise Work in process inventory", "Maximising machine or worker utilization", "Reduce setup times" and "Minimising production and worker costs". Cost-quality trade-off determination is very crucial in a competitive world. The ultimate objective results from these objectives is increasing profitability.

There are some conflicting objectives like the WIP inventory reduction may lead to an increase in worker idle time. The accurate scheduling technique depends on order size, operation contents and sequences, complexity of jobs and the relative importance placed on these objectives. Identifying a solution in real time is advisable considering tradeoff between the resources spent in identifying an optimal solution and the time taken for it. (Heywood et al., 1997).

1.4.1 Single and Multi-Objective Optimization in FMS

A schedule focused on minimizing a single performance metric such as makespan, flow time, tardiness, idle time, or the number of tardy jobs is classified as single-objective optimization. These objectives are generally independent of each other.

In contrast, multi-objective optimization involves minimizing multiple criteria simultaneously to optimize the overall production cost and efficiencyIt evaluates trade-offs between objectives and frequently results in a set of Pareto-optimal solutions instead of a single optimal outcome. A Pareto

solution set provides multiple alternatives for decision-makers, enabling the selection of the most suitable schedule based on operational priorities. Recently, energy consumption has gained prominence as a critical objective in manufacturing optimization, aiming to reduce the environmental impact and operational costs. Energy-efficient scheduling strategies seek to minimize power consumption during machine idle times and optimize tool usage to lower energy demands, aligning with modern sustainability goals (Li et al., 2023; Zhang & Liu, 2022).

1.4.2 Tool Scheduling in Flexible Manufacturing Systems (FMS)

In modern FMS, tools required for machining operations are often delivered from a Central Tool Management (CTM) system using automated tool transporters. When multiple machines request the same tool, the machine with the highest priority is served first, while others remain idle, resulting in "tool waiting time." Effective tool scheduling strategies can minimize this idle time and reduce tooling costs by optimizing tool assignment and sharing across the system. Advanced approaches, such as real-time tool monitoring and predictive analytics, further enhance tool availability and reduce delays.

1.4.3 Automated Guided Vehicle (AGV) Scheduling in FMS

Automated Guided Vehicles (AGVs) are autonomous vehicles used for material handling in FMS that follow specified pathways. AGV activities include two types of trips:

Loaded Trip: The AGV carries a job between machines.

Empty Trip: The AGV moves without carrying a load, often returning after a job drop-off.

To minimize total transportation time and reduce empty trips, effective AGV scheduling is essential. Advanced routing algorithms and adaptive scheduling approaches have been developed to optimize AGV movements, ensuring minimal energy consumption and reduced operational delays (Wang et al., 2023).

1.4.4 Simultaneous Scheduling in FMS

To enhance FMS performance, machines and additional resources like tools, AGVs, AS/RS, and robots can be scheduled together. In simultaneous scheduling of machines and tools, an operation cannot commence until the necessary tool arrives from the CTM or another machine, leading to increased completion times due to machine idleness. This strategy becomes crucial when tools are limited or expensive, requiring careful resource coordination.

When extending simultaneous scheduling to include AGVs, the complexity increases, as operations can only proceed when both tools and jobs are delivered to the workstation. This often results in significant idle time due to resource dependency. However, synchronized scheduling and real-time decision-making models can significantly reduce idle time and improve system efficiency. Energy-aware simultaneous scheduling models are now being developed to ensure energy-efficient operations alongside productivity

improvements (Zhou et al., 2023).

1.5 Challenges and Opportunities in Indian Manufacturing

The Indian manufacturing industry plays a pivotal role in the nation's economic growth, with Micro, Small, and Medium Enterprises (MSMEs) forming its backbone. India has about 6.3 crore MSMEs (Ministry of MSME, 2023), contributing nearly 30% to the GDP and 48% of exports. Most of these enterprises operate through job-shop style production systems, such as machine shops, fabrication units, and auto component manufacturing, which often struggle with tardiness, bottlenecks, and limited automation. At a broader level, manufacturing contributes around 17–18% to India's GDP (Economic Survey 2023–24), and under the Government's *Make in India* initiative, this share is targeted to reach 25% by 2025, making efficient scheduling and productivity improvement crucial.

Energy consumption adds another dimension, with industry accounting for nearly 40% of India's total energy use (IEA, 2022), dominated by energy-intensive sectors like steel, cement, aluminium, and textiles. National initiatives such as the PAT scheme by the Bureau of Energy Efficiency have set ambitious goals of 5–10% energy savings per cycle, and optimization-based approaches like NvPSO, which demonstrate around 9% energy savings, align closely with these targets.

At the same time, India's adoption of automation and Industry 4.0 technologies remains modest, with less than 15% of factories being highly automated compared to ~65% in advanced economies (IMTMA). However, opportunities are emerging in areas such as Automated Guided Vehicles (AGVs), with the Indian market projected to grow at over 10% CAGR between 2023 and 2030, especially in automotive, e-commerce, and electronics clusters like Pune, Chennai, Gurugram-Manesar, Bengaluru, and Coimbatore. Persistent inefficiencies such as 20–30% machine idle time (CII reports), order tardiness penalties of 5–8% in export-oriented units, and avoidable energy wastage of nearly 8–10% (TERI, 2021) highlight the pressing need for intelligent scheduling and dynamic optimization solutions. Against this backdrop, the present research addresses these challenges by proposing advanced scheduling strategies that integrate optimization algorithms with automation, positioning them as highly relevant for India's manufacturing transformation.

1.6 Problem Definition

The goal in an FMS environment is to schedule machines, material handling systems (AGVs), and tools all at the same time, while accounting for work transfer times between the load/unload (L/U) station and machines, as well as among machines. Furthermore, tool transfer times between the Central Tool Management (CTM) system and the machines, as well as between machines, must be considered. The primary goal is to minimise the makespan, which is defined as the total time elapsed between retrieving the first work from the L/U station and completing the last operation. An additional objective

involves minimizing **energy consumption** by reducing idle machine times, unnecessary tool movements, and optimizing AGV routes to achieve sustainable manufacturing operations.

1.6.1 Objectives of the Study

The study aims to develop algorithms for makespan and energy optimization in simultaneous scheduling of the following FMS components:

- **Tools and Machines:** Consider tool transfer times between CTM and machines, as well as between machines themselves.
- **Machines, AGVs, and Tools:** Without considering tool transfer times between CTM and machines.
- **Machines, AGVs, and Tools:** Taking into account tool transfer times between CTM and machines, as well as between machines themselves**.**

The research also focuses on the application of advanced **nature-inspired metaheuristic algorithms** to minimize both makespan and energy consumption for the concurrent scheduling of:

- Machines & AGVs.
- Machines and Tools (without considering tool transfer times between CTM and machines, and among machines).
- Machines and Tools (considering tool transfer times between CTM and machines, and among machines).
- Machines, AGVs, and Tools (without considering tool transfer times between CTM and machines, and among machines).
- Machines, AGVs, and Tools (considering tool transfer times between CTM and machines, and as well as between machines themselves).

1.6.2 Key Objectives:

1. **Minimization of Makespan:** Optimise the scheduling process to reduce the total production time.
2. **Minimization of Energy Consumption:** Reduce energy usage by minimizing idle machine time, optimizing AGV routes, and limiting redundant tool movements.
3. **Flow Analysis:** Evaluate the influence of both **job flow** and **tool flow** on makespan and energy efficiency of the FMS.
4. **Algorithmic Efficiency:** Identify the most effective metaheuristic algorithms for simultaneous

scheduling in FMS, focusing on both productivity (makespan) and sustainability (energy consumption).

1.7.Outline of the thesis

Chapter 1 – This first chapter offers a complete introduction of the research issue. It summarises the problem statement, defines the study's goals and objectives, and highlights the major contributions produced by this research.

Chapter 2 – This chapter provides a detailed review of the available literature relevant to the subject.It explores past research efforts, identifies relevant findings, and highlights research gaps that this work aims to address and build upon.

Chapter 3 – This section introduces metaheuristic optimisation techniques and proposes three novel metaheuristic approaches tailored to address different scheduling problems.

Chapter 4 – Focuses on the formulation of the Concurrent Scheduling Problem with multiple objectives. It includes a performance analysis of the proposed algorithms, comparing them with other metaheuristic methods and evaluating their effectiveness against each other.

Chapter 5 – Examines the optimisation of concurrent scheduling using a combined objective approach. It assesses the performance of the proposed solutions by benchmarking them against existing techniques and comparing their outcomes.

Chapter 6 – This chapter explores the concurrent scheduling of m/cs, tools, and automated guided vehicles (AGVs) using multi-objective optimisation strategies with an emphasis on reducing energy consumption.

Chapter 7: In this chapter, a case study is presented to demonstrate the application of the proposed algorithm in the Indian manufacturing industry.

Chapter 8 – The final chapter presents the assumptions drawn from the learning, summarising the findings discussed in previous chapters. It also outlines possible directions and opportunities for future research.

1.8 The Contributions of the Present Work

This work contributes to field of Flexible Manufacturing Systems (FMS) by developing advanced algorithms for concurrent scheduling of m/c’s, material handling systems (AGVs), and tools. The

primary contributions of this research include:

- Development of energy-aware simultaneous scheduling models that minimize makespan and energy consumption.
- A hybrid metaheuristic algorithm is employed to overcome certain limitations of individual optimization techniques. This hybrid approach integrates swarm intelligence with population-based methods, incorporating adaptive parameter tuning. Three distinct algorithms are developed to improve tool scheduling efficiency and minimize machine idle time.
- Additionally, the study will incorporate case studies for evaluation. These case studies will involve diverse machine types and a range of job assignments. Each scenario will include multiple machines and multiple jobs to reflect realistic and complex scheduling environments.
- Implementation of AGV scheduling models aimed at minimizing transportation time and empty trips.
- Exploration of synchronized scheduling approaches for combined machine, tool, and AGV coordination.
- Evaluation of the impact of job and tool flow on production efficiency and sustainability.

These contributions provide a foundation for improving both productivity and sustainability in modern FMS environments.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Flexible Manufacturing Systems (FMS) have become pivotal in modern manufacturing, enabling the production of diverse products without extensive reconfiguration. This flexibility enhances adaptability in dynamic market conditions. Generally, flexibility is regarded as an interfacing character between a system and its peripherals.

FMS scheduling involves several key elements, including machine loading, part routing, task scheduling for manufacturing, tool planning and distribution, formulating a strategy for utilizing buffers and planning for AGV operations Numerous studies suggest that the processing time and cost of a job, as well as the setup time and cost associated with it, are often overlooked. While this simplification may ease the analysis and suit specific applications, it significantly compromises the quality of solutions for scheduling scenarios where precise consideration of setup time and cost is crucial.

Including setup time and cost in scheduling, design became a key consideration during the mid-1970s. Several surveys have been conducted on this topic, with their findings presented for discussion and analysis. Scheduling problems are often categorized as either static or deterministic, with research focusing on both models .(Cheng et al., 2000)conducted an extensive review of scheduling challenges in flow shops, while (Potts & Kovalyov, 2000)provided insights from their survey on scheduling-related to batching applications. (Allahverdi et al., 2008) reviewed scheduling problems under stochastic and dynamic conditions across various shop environments. These environments included single-machine scheduling, parallel machine groups, job shops, and flow shops, among others.

The results of these surveys have guided subsequent research efforts, emphasizing the importance of including setup time and cost as critical factors in scheduling. This focus has gained significant traction due to its impact on cost savings and profit enhancement, particularly in real-time applications

This chapter aims to present an overview of the current state of research on Flexible Manufacturing Systems (FMS), with a focus on its design, control, scheduling, optimisation, and performance evaluation. It begins by introducing the concept of FMS and its key

components, followed by a review of the literature on FMS design. The discussion then transitions to FMS scheduling, covering the determination of task order and timing for system components. Various scheduling techniques, including rule-based and optimisation-based approaches, and the factors influencing their selection, will be explored.

Lastly, the chapter will examine FMS optimisation, which aims to enhance system performance by efficiently utilising resources like raw materials, energy, and labour. The review will highlight different optimization techniques developed for FMS and the criteria influencing their adoption.

2.2 Overview of flexible manufacturing system

Flexible Manufacturing Systems (FMS) enable the production of a wide variability of product types without requiring reconfiguration across the entire manufacturing line. These systems consist of numerous numerically controlled machines with multifunctional capabilities [38]–[42], automated material handling systems [43]–[45], and an integrated online computer network [46] capable of managing and directing the entire operation [47].

FMS is highly valued in modern manufacturing due to its ability to deliver rapid turnaround times, minimize labour and inventory costs, and maintain superior product quality [48]. To enhance the global efficiency of FMS, manufacturing activities, including transportation and storage tasks, must be meticulously scheduled. Scheduling is inclined by factors like the automation level, specific FMS characteristics, the plant's location, operational rules, and available resources [49].

Developing high-quality scheduling that integrates all FMS resources—such as machines, tools, automated guided vehicles (AGVs) [50], and buffers—is a critical operational challenge in these environments [51]. Effective scheduling is essential for optimizing resource utilization and improving overall system performance [52]. The dynamic nature of incoming jobs, with frequent changes in part designs, adds complexity to the scheduling process. The focus is on effectively scheduling jobs to maximize system utilization and output, especially when machines are equipped with diverse tools and tool maintenance schedules.

In this context, scheduling performance is typically evaluated using metrics such as production time, delays, and mid-term operational efficiency. These factors contribute to the complexity of scheduling, necessitating robust frameworks for managing FMS. In engineering and manufacturing, scheduling is a critical tool that significantly impacts process productivity [53]. The primary goals of scheduling are to minimize production time and costs while determining the specific facility, equipment, and personnel required for manufacturing [54]. The overarching objective of production scheduling is to enhance operational efficiency while reducing costs [55].

Various software solutions are available to model manufacturing processes, covering everything from raw material procurement to production and final product delivery. For instance, raw materials like “Steel-M1” and “Gear-X” are used to manufacture products such as AX-100, AX-200, BX-100, BX-200, CX-100, and CX-200. The associated operations include cutting, additional processing, assembly, packaging, and distribution [56].

2.3 Review of Literature on FMS: Problem Types and Optimisation Strategies

2.3.1 Research on the Design and Implementation of Flexible Manufacturing Systems (FMS)

(Stecke, 1985) provided a thorough analysis of the key challenges in designing, planning, scheduling, and controlling Flexible Manufacturing Systems, establishing a strong foundation for future FMS research and implementation. He et al. explored the sequencing of parts and scheduling processes within a mass customization environment, focusing on the coordination of robots and machines through specific scheduling rules. (Negahban & Smith, 2014) carried out an in-depth review addressing both design and operational aspects of FMS, highlighting the use of various simulation tools applied in related studies. (Freitag & Hildebrandt, 2016) proposed a simulation-based approach to a semiconductor manufacturing scenario, employing a multi-objective genetic algorithm to optimize due-date-related performance metrics

(Kouvelis, 1992) provide a comprehensive overview of state-of-the-art research on FMS design and planning issues. Their review primarily focuses on research outcomes from the FMS literature that assist managers in establishing highly efficient manufacturing systems. They emphasize combinatorial optimization approaches to FMS planning problems and include research contributions made after 1986, which were previously excluded from earlier survey papers. This study addresses critical operational research models for FMS, enabling better decision-making in system setup and management.

(Bilge & Ulusoy, 1995) introduced an offline methodology for simultaneously configuring equipment and handling systems in a Flexible Manufacturing System (FMS) with the objective of minimizing makespan. They employed a descending time window method to address a nonlinear mixed-integer programming model. Similarly, Ulusoy et al. (1997) utilized a Genetic Algorithm (GA) approach to explore the same issue, encoding process numbers and Automated Guided Vehicle (AGV) tasks in chromosomes, which required the development of specialized genetic parameters.(LEE, 1999) explores the design of FMS from a cost-effectiveness perspective. The study utilizes closed queuing network models to optimize design decisions such as the number of machine groups, machines per group, workload allocation, pallet

numbers, transporters, and batch sizes. Lee highlights the interdependence of these factors and presents both optimal and heuristic methods to resolve the complexity of design. The heuristic methods, in particular, are recognized for their efficiency in handling large-scale systems with multiple machine types, providing practical solutions for time-constrained scenarios.

(MacCarthy & Liu, 1993) applied a Support Vector Machine (SVM) approach for FMS scheduling, focusing on dispatching rules under dynamic conditions such as part variation and job influx. Using a radial basis function kernel, the SVM model outperformed traditional models and required minimal setup time.

According to (Jerald et al., 2005) many of these techniques are tailored to specific goals or problem scenarios, considering factors such as computational time and genetic operators.Han et al., 2018 proposed a heuristic scheduling approach using T-timed Petri nets for flexible manufacturing. Their heuristic functions effectively reduced makespan while balancing the number of states and computational time, proving superior to traditional methods through experimental validation.(Zambrano Rey et al., 2014) designed a semi-hierarchical optimization architecture for modern FMS. Their approach balanced local and global optimization using simulation-based methods, focusing on minimizing completion time variance. Though effective for production control, the study lacked a comprehensive mathematical foundation for the proposed method.

Their optimization approach considered both makespan and tardiness functions. Nageswara rao et al. (2015) implemented a hybrid meta-heuristic algorithm to optimize job sequencing based on makespan reduction while minimizing delivery costs and boosting productivity. Erdin and Atmaca (2015) developed an analytical model using bottleneck principles and clustering for effective workstation utilization and part sequencing. Their model was compared with traditional manufacturing models to evaluate efficiency. Baruwa and Piera (2015) introduced a Coloured Petri Net (CPN) for optimal scheduling in FMS. Their method used reachability graphs and structural equivalence graphs to reduce memory usage while maintaining search efficiency for large-scale FMS problems.

(Erdin & Atmaca, 2015)focuses on the practical implementation of FMS design, specifically in calculating the necessary number, utilization, and sequence of workstations and plant layouts for given production requirements. Analytical bottleneck models and rank order clustering techniques are applied to optimize the system. The study simplifies analysis through the creation of manufacturing cells for similar parts, ensuring the effective use of workstations. The results demonstrate the superiority of FMS over conventional manufacturing conditions, underscoring the effectiveness of systematic design methodologies.

Krishna et al. (2016) investigated complex scheduling problems in FMS, emphasizing material

handling alongside machine scheduling. They introduced a FlexSim-based simulation model to assess the impact of buffer management and parallel scheduling on makespan reduction. Gang and Quan(2016) proposed a multi-layered FMS scheduling model with a list-based algorithm to handle varying levels of flexibility, which was validated through a planning layout for accuracy and adaptability.

Mallikarjuna et al. (2017) employed a multi-objective optimisation technique for FMS scheduling, using metaheuristics like Simulated Annealing (SA) and Particle Swarm Optimisation (PSO). Their study showed SA yielded better results than PSO, although both methods had similar computational times Akhtar et al. (2018) introduced a batch size optimisation mechanism, accounting for completion time, lateness penalties, and setup durations. Malik and Pena (2018) explored task scheduling in FMS using model checking techniques, treating it as a discrete event system with finite state machines and restricted safety behaviours. Supremica, a model-checking tool, was employed to compute optimal scheduling times.

(Manu et al., 2018) . provides a review that encapsulates the various aspects of FMS, highlighting its critical role in modern manufacturing. FMS integrates components such as computer-programmed machine tools, automated material handling systems, robots, and self-diagnostic facilities into a cohesive production system. Manu emphasizes the significant investment required for FMS and its potential as a competitive tool in manufacturing. The review bridges gaps between critical methodological aspects and highlights the importance of research in optimising FMS implementation. The study underscores the need for continuous exploration of FMS to achieve greater operational flexibility and efficiency.

Bathini et al. (2019) focused on minimizing machine idle time and penalty costs with equal priority, developing a Combined Objective Function (COF) to address both. Lee and Ha (2019) developed a genetic algorithm for integrated process planning and scheduling in FMS, addressing combinatorial optimization with NP-complete complexity. Their approach integrated multiple flexibilities into a single chromosome, simplifying genetic operators and improving makespan by approximately 17% while reducing computational time.

Karunagaran et al. (2022) combined two algorithms with simulated annealing for effective local search. Their experiments revealed that the Hybrid Adaptive Firefly Algorithm (HAdFA) outperformed both the Hybrid Firefly Algorithm (HFPA) and other metaheuristics by enhancing convergence speed and enabling the discovery of multiple optimal solutions.

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

Flexible Job Shop Scheduling Problems (FJSSP) have been widely studied due to their complexity and practical importance in manufacturing systems. Various optimization techniques and hybrid algorithms have been proposed to improve efficiency in scheduling machines, tools, and material handling systems while minimizing objectives such as makespan, tardiness, and energy consumption.

Mastrolilli and Gambardella [2000] emphasized the effectiveness of neighbourhood functions in enhancing solution quality for the flexible job shop scheduling problem. They highlighted the importance of selecting a suitable neighbourhood function and developed metaheuristic algorithms such as simulated annealing and tabu search.

Mati et al. [2001] proposed a heuristic approach using an integrated greedy algorithm combining priority rules for job sequencing and dispatching rules for machine assignment. Their work focused on reducing makespan by prioritizing job sequences and assigning jobs based on machine availability.

Xia and Wu [2005] introduced a hybrid optimisation strategy combining a local search algorithm and a multi-objective genetic algorithm (MOGA). The local search refined the MOGA solutions, improving efficiency and quality in balancing multiple objectives.

Motaghedi-Larijani et al. [2010] utilized a multi-objective genetic algorithm (MOGA) for simultaneous optimisation of makespan and tardiness. They introduced a heuristic initialization technique and a novel fitness function integrating both objectives.

Xing et al. [2010] developed a knowledge-based ant colony optimization (KBACO) algorithm where prior optimization data guided current heuristic searches, effectively improving solution efficiency.

Khadwilard et al. [2012] implemented the Firefly Algorithm and explored the impact of different parameter settings like light absorption and attraction coefficients on the performance of FJSSP scheduling. Bhushan and Kumar [2014] employed the Taguchi philosophy combined with genetic algorithms to optimise job scheduling in dynamic FJSSP scenarios, focusing on increasing throughput and machine utilization. Karthikeyan et al. [2012] proposed a hybrid strategy combining data mining techniques with particle swarm optimization (PSO) to extract near-optimal solutions for complex job shop scheduling problems.

Yanibelli and Amandi [2013] developed a hybrid optimization combining multi-objective evolutionary algorithms with simulated annealing to balance project duration and resource allocation efficiency. Roshanaei et al. [2013] proposed a metaheuristic combining artificial immune and simulated annealing (AISA) algorithms, validated through a mould and die shop case study.

Yuan et al. [2013] developed a hybrid harmony search algorithm (HHS) with a novel two-vector encoding method, effectively reducing search space and improving scheduling efficiency.

Buddala and Mahapatra [2019] applied teaching-learning-based optimization (TLBO) with a new local search inspired by genetic algorithms to minimize makespan for benchmark FJSSP problems.

Li et al. [2020] introduced an adaptive evolutionary algorithm with novel encoding techniques and parameter adjustments, which significantly improved scheduling outcomes in FJSSP. Ning et al. [2021] developed a quantum bacterial foraging optimization (QBFO) algorithm focusing on low carbon emissions in FJSSP and validated results using ANOVA. Jiang et al. [2022] applied a discrete animal migration optimization (DAMO) approach for dual-resource constrained, energy-efficient scheduling, minimizing total energy consumption.

Long et al. [2022] proposed a dynamic self-learning artificial bee colony (DSLABC) approach combining Q-learning and the ABC method for flexible job shop scheduling, focusing on the dynamic addition of jobs during the scheduling process.

Philipp Schworm et al. (2023) introduce a Quantum Annealing-based algorithm (QASA) to address FJSS problems, optimising multiple criteria such as makespan, total workload, and job priority. The approach combines quantum annealing with classical techniques and employs problem decomposition for large instances. Experimental results demonstrate that QASA outperforms classical algorithms in solution quality.

Imanol Echeverria et al. (2023) propose a deep reinforcement learning (DRL) method utilising heterogeneous graph neural networks to solve large FJSS instances. The approach captures complex relationships between operations and machines, leading to improved decision-making. Experiments on benchmarks show that this method outperforms traditional dispatching rules and state-of-the-art DRL methods, especially for large instances.

Runqing Wang et al. (2023) presents an end-to-end learning framework combining self-attention models and DRL. A dual-attention network captures intricate relationships between operations and machines, enhancing feature extraction for high-quality decision-making. The proposed method outperforms traditional priority dispatching rules and state-of-the-art DRL methods on synthetic and benchmark datasets.

Hessam Bakhshi-Khaniki et al. (2024) addresses FJSS involving reconfigurable machine tools with configuration-dependent setup times, integrating human factors like worker assignments and rest periods, and aiming to minimize energy consumption. A mixed-integer programming model and a memetic algorithm are developed. Comparisons indicate the memetic algorithm's efficiency, especially for larger problem instances.

Wenbo Chen et al. (2023) proposed a two-stage learning framework (2SL-FJSP) that models the hierarchical nature of FJSS decisions. It uses a confidence-aware branching scheme and a novel symmetry-breaking formulation to improve learnability. Evaluations on benchmark instances show that 2SL-FJSP generates high-quality solutions rapidly, outperforming state-of-the-art reinforcement learning approaches and common heuristics.

The reviewed literature highlights a trend towards hybrid and nature-inspired metaheuristic algorithms for solving FJSSP. These algorithms often combine multiple strategies to balance solution quality and computational efficiency. Modern approaches also emphasize energy consumption and sustainability along with traditional objectives like makespan and tardiness, aligning with industry demands for efficient and eco-friendly manufacturing solutions.-doubtful it is required or not

2.3.3 Studies related to Scheduling of AGVs and machines in FMS

Researchers have explored various algorithms and methodologies to optimize scheduling and resource allocation in Flexible Manufacturing Systems (FMS) and related environments. This summary highlights key contributions and advancements in the field, emphasizing scheduling of machines, tools, and Automated Guided Vehicles (AGVs) for enhanced system performance.(Bilge & Ulusoy, 1995) implemented a Genetic Algorithm (GA) to concurrently schedule machines and AGVs in an FMS equipped with multiple machining centers and identical AGVs. Their goal was to minimize the makespan, effectively enhancing operational efficiency.(Keung et al., 2001)examined machine allocation and scheduling in parallel workstations with shared tools. They utilized a GA to reduce tool-switching instances, thereby optimizing workflow and minimizing tool management overheads.

(Ecker & Gupta, 2005) proposed an algorithm to sequence tasks on a Single Flexible Machine (SFM) with a tool magazine. Their approach aimed at minimizing tool changes, reducing setup times, and improving machine utilization. Karzan and Azizoğlu developed a Branch-and-Bound algorithm coupled with a beam search technique to address the issue of minimizing tool transporter movements in single versatile machine setups with limited tool magazine capacity. (Kumar N & Sridharan, 2007)focused on tool-sharing and scheduling challenges in FMS. Their research sought to minimize metrics such as mean tardiness, mean flow time, and conditional mean tardiness, contributing to improved timeliness and resource utilization.Kim et al. introduced a hybrid scheduling methodology combining inductive and competitive neutral strategies to design multi-objective FMS schedules. This integrated approach addressed various performance metrics simultaneously.(Udhayakumar & Kumanan, 2010) utilized Ant Colony Optimization (ACO) to determine optimal job and tool sequences, aiming to minimize the makespan. Their approach demonstrated the potential of bio-inspired algorithms in FMS optimization. In another study,(Udhayakumar & Kumanan, 2012) employed several

algorithms, including Particle Swarm Optimization (PSO), Simulated Annealing (SA), ACO, and GA, to address tardiness minimization in FMS. The comparative analysis highlighted the strengths and limitations of each method.

(Raj et al., 2014)proposed revised heuristics, algorithms, and an Artificial Immune System (AIS) for machine and tool scheduling, specifically targeting makespan reduction. Their work showcased innovative strategies to enhance FMS scheduling efficiency. (Nageswararao et al., 2015) introduced a Hybrid Genetic Vehicle Heuristic Algorithm (HGVHA) to tackle the simultaneous scheduling of machines and AGVs. This approach aimed to minimize both the makespan and mean tardiness, emphasizing the role of AGV optimization in improving FMS productivity.

(Özpeynirci, 2015) developed a mathematical model incorporating time indices to optimize machine and tool scheduling in FMS. The primary focus was on minimizing the makespan, providing a robust framework for system-level optimization.(Pena et al., 2016) combined a hybrid Genetic Algorithm (GA) with an improved local search technique to schedule tasks across multiple parallel machines. Their approach accounted for tool wear and the necessity of tool changes, resulting in more effective tool management. (Reddy et al., 2018) utilized a Cuckoo Search Algorithm (CSA) for simultaneous machine and tool scheduling in a Multi-Machine FMS (MMFMS), with the objective of minimizing makespan.In another study, .(Mareddy et al., 2022). (b) employed nature-inspired algorithms, including the Flower Pollination Algorithm (FPA), for scheduling tasks in MMFMS. The FPA demonstrated superior performance in terms of optimizing system efficiency.

Beezão et al. addressed the challenge of optimizing makespan in parallel machines under tooling constraints. They implemented an adaptive large neighborhood search metaheuristic to tackle this complex problem effectively.(Baykasoglu & Ozsoydan, 2017)focused on issues related to automatic tool changer indexing and tool changes in automated machining centers. They applied a simulated annealing algorithm to address these challenges, enhancing the efficiency of tool-changing operations.

(Paiva & Carvalho, 2017) explored job sequencing and tool-changing problems using graph representation techniques combined with heuristic and local search methods. Their work provided a structured approach to solving complex scheduling issues.(Gökgür et al., 2017) employed constraint programming models to optimize tool scheduling and allocation in parallel machine environments with predefined tool quantities. Their primary objective was to minimize the makespan, contributing to streamlined production processes.

Pandey and Singh conducted a review on the design and control of automated guided vehicles (AGVs). Their study integrates various strands of AGV research, focusing on critical challenges like vehicle

scheduling. The review addresses scheduling issues in different job shop setups, such as vehicle dispatching, guide-path design, and routing.

Chawla et al. developed the Modified Memetic Particle Swarm Optimization Algorithm (MMPSO) by combining Particle Swarm Optimization (PSO) and Memetic Algorithm (MA). MMPSO is designed to optimize the scheduling of multi-load AGVs by minimizing travel and waiting times. It balances global search (PSO) and local search (MA) methods, yielding efficient initial solutions for scheduling problems.

Reddy et al. addressed the simultaneous scheduling of AGVs, machines, and tools using a flower pollination algorithm (FPA) and a nonlinear mixed integer programming (MIP) model. Zheng et al. minimized makespan by scheduling machines and AGVs together using a Tabu search technique, introducing two-dimensional solution representation and improved lower bound computations for larger problems.

Prasad and Rao proposed a black widow optimization algorithm for simultaneous machine and tool scheduling in a flexible manufacturing system (FMS). Their approach aimed at minimizing production time and makespan. Sreenivas et al. used a simulated annealing algorithm for parallel scheduling of AGVs and machines, focusing on makespan, mean makespan, and tardiness reduction, though the mathematical derivation was minimal. Saren et al. investigated decision-making strategies for FMS through hierarchical modeling. Using CPN Tools software, their analysis focused on the arrival and processing times of parts in a flexible manufacturing cell (FMC) laboratory setup, but lacked a conclusive mathematical derivation.

Gothwal and Raj developed a proactive task management framework for FMS, integrating simple additive weighting and weighted product scheme with analytical hierarchy process (AHP). They used fuzzy logic for converting qualitative data to quantitative measures.Chawla et al. also studied dynamic job selection for multi-load AGVs, emphasizing dispatching rules and vehicle speed, but lacked mathematical formulation for scheduling analysis.

Erol et al. proposed a multi-agent system for machine and AGV scheduling, emphasizing negotiation-based scheduling strategies. Burnwal and Deb applied a cuckoo search algorithm for FMS scheduling, focusing on reducing penalties and improving machine utilization using a modified Levy flight operator.Huang et al. proposed a P-timed Petri net model for FMS scheduling, integrating heuristic functions for performance improvement. Başak and Albayrak developed a Petri net model for FMS, using object-oriented Petri nets (OOPNs) for effective production control, specifically tested in Valeo Turkey.Chawla et al. further investigated dispatching rules for multi-load AGVs in FMS, showing throughput variations based on AGV speed and fleet size without detailed mathematical analysis.

Kumar et al. explored meta-heuristic approaches like Bacterial Foraging Optimization Algorithm (BFOA), Genetic Algorithm (GA), and Differential Evolution (DE) for FMS scheduling, using Pro Model software for analysis.Mousavi et al. combined genetic algorithms, particle swarm optimization, and a hybrid GA-PSO approach for AGV scheduling, focusing on makespan reduction and battery charging considerations, validated using Flexsim software.

Mishra et al. used PSO for multi-objective scheduling in FMS, addressing transport and storage tasks but lacking detailed job scheduling considerations. Kamatchi and Saravanan proposed a modified firefly algorithm for open shop scheduling, suggesting future work on multi-objective functions and fuzzy models.Several researchers, including Rashmi and Bansal, applied ant colony optimization (ACO) for AGV task scheduling, showing dynamic performance improvements. Mehrabian et al. presented a two-objective fuzzy mathematical programming model for AGV routing and scheduling using NSGA-II and PSO but lacked practical validation.

Udhayakumar and Kumanan used ACO with Giffler extension and Thompson algorithms for active job scheduling in FMS. Mathew and Saravanan developed a genetic algorithm for modern FMS, focusing on reducing machine idling and penalty costs.Kumar et al. validated genetic and differential evolution algorithms for FMS scheduling using MATLAB with 16 machines and 43 jobs. Nidhiry and Saravanan applied NSGA-II for CNC machine scheduling, focusing on minimal idling and penalty costs.

Marichelvam et al. proposed a hybrid monkey search algorithm for flow shop scheduling, minimizing makespan and total flow time through combinatorial optimization techniques.Amirteimoori et al. (2023) developed a Parallel Two-Step Decomposition-Based Heuristic (PTSDBH) for the concurrent scheduling of jobs and AGVs in hybrid job shop systems. A key contribution was its focus on conflict-free AGV routing, an aspect often neglected in similar studies. Leveraging parallel computing, PTSDBH efficiently managed large-scale scheduling problems and outperformed established metaheuristics such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO) in both speed and solution quality. Its emphasis on preventing AGV collisions further improved system reliability. Zou et al. (2023) proposed a novel approach for the less-explored Multi-AGV Scheduling Problem with Maintenance (MAGVSCM) in manufacturing workshops. Their self-adaptive iterated greedy algorithm (SAIG) effectively minimized overall costs, including travel, penalty, and vehicle expenses. Key innovations, such as an advanced solution representation, adaptive strategies, and refined heuristics, contributed to the SAIG's superior performance compared to existing methods, as validated through extensive experimental testing.

2.3.4 Studies on Particle swarm optimisation and genetic algorithm on other domains

Numerous studies have explored the applications of Particle Swarm Optimization (PSO) and Genetic Algorithms (GA) across different domains, enhancing optimization techniques through novel adaptations and hybrid approaches.

Garg, Shukla, and Tiwari (2019) proposed an Adaptive Exploration Robotic PSO (AERPSO) for autonomous robots to improve multi-objective search efficiency. Their method demonstrated a 40% reduction in search time and a 25% improvement in detection rate by enhancing exploration capabilities and obstacle avoidance. Wu et al. (2020) developed an innovative PSO variant that integrates the Surprisingly Popular Algorithm (SPA) with fitness evaluation, enabling better performance in large-scale problems by balancing exploration and exploitation. Experimental validation confirmed its superiority over conventional PSO techniques in diverse optimization tasks.

Qu et al. (2021) introduced Explicit Scheme Adaptive PSO (ESAPSO), a feature selection algorithm utilizing an explicit representation approach for particle encoding. Unlike implicit methods used in high-dimensional datasets, ESAPSO significantly reduced computational complexity and memory demands while enhancing classification accuracy. This was achieved through feature grouping strategies and a size-adaptive expansion mechanism, which produced feature subsets of comparable sizes. Amirteimoori et al. (2022) developed a Mixed-Integer Linear Programming (MILP) model for optimizing job and transporter scheduling in flexible flow shop environments. Their research employed a Parallel PSO-GA Algorithm (PPSOGA), demonstrating enhanced solution quality and computational efficiency, highlighting the advantages of parallel processing in optimization problems.

Wang et al. (2022) presented a novel Multi-State Scheduling Algorithm (MSSA) for Automated Guided Vehicles (AGVs) in Flexible Manufacturing Systems (FMS). MSSA incorporated neural network-based travel time prediction, optimizing AGV operations and reducing makespan. Simulation results confirmed improvements in task execution time, AGV utilization, and overall system efficiency. In a related study, an improved Artificial Bee Colony (ABC) algorithm, IGAL-ABC, was introduced, effectively balancing exploration and exploitation, leading to superior convergence and solution quality compared to traditional ABC variants. Yang et al. (2021) developed ACoM-ABC, an advanced ABC algorithm integrating an adaptive covariance matrix, thereby reducing reliance on natural coordinates and improving exploitation. Benchmark experiments demonstrated ACoM-ABC's ability to outperform six ABC and six evolutionary algorithms, with practical applications in hearing loss detection.

Chen et al. (2023) addressed the challenge of water quality index (WQI) prediction in river pollution management. They introduced the Adaptive Evolutionary Artificial Bee Colony (AEABC) algorithm,

which incorporated Back Propagation Neural Networks (BPNN) and dynamic adaptive factors. This method achieved notable convergence in 14 iterations, reducing errors by up to 25.2%, proving highly effective for environmental monitoring. Similarly, Chaudhary (2023) proposed MABC-SS, a modified ABC algorithm for global optimization. Enhancing population initialization and employing an advanced rate of change selection strategy, MABC-SS outperformed conventional ABC variants in benchmark evaluations, achieving optimal or near-optimal results.

Several studies have explored the integration of PSO and GA with scheduling optimization. Garg et al. (2022) refined AERPSO for multi-target robotic exploration, demonstrating enhanced search efficiency. Wu et al. (2023) introduced a hybrid PSO-SPA model combined with adaptive Euclidean distance-based topology, optimizing large-scale heterogeneous population problems. Amirteimoori et al. (2022) further applied a PSO-GA hybrid approach to improve scheduling in flexible flow shop environments, benefiting from parallel computing.

Other researchers have focused on energy-aware scheduling in manufacturing. Bruzzone et al. (2012) developed an Energy-Aware Scheduling (EAS) framework, optimizing energy consumption while maintaining job sequences. Dai et al. (2019) formulated a multi-objective optimization model for flexible job shop scheduling, reducing both makespan and energy consumption using an enhanced genetic algorithm. Fang and Lin (2013) addressed scheduling in multi-machine systems with adjustable processing speeds, minimizing job tardiness and energy use through heuristic and PSO-based techniques. Gahm et al. (2016) proposed an Energy-Efficient Scheduling (EES) framework, encompassing energy supply, demand, and consumption management.

Recent advancements in metaheuristic algorithms have also contributed to optimization strategies. Trojovský & Dehghani (2023) introduced the Walrus Optimization Algorithm (WaOA), which incorporated exploration, migration, and exploitation phases, demonstrating robust performance in benchmark tests. Fahmy et al. (2024) applied WaOA to lithium-ion battery (LIB) modeling, optimizing resistance-capacitance (RC) networks for accurate voltage estimation. Han et al. (2024) further validated the Walrus Optimizer (WO) on standard benchmarks and engineering problems, confirming its reliability and stability. Hasanien et al. (2024) proposed the Enhanced Walrus Optimization (EWO) algorithm for solving the Probabilistic Optimal Power Flow (POPF) problem in power grids, optimizing cost efficiency while integrating renewable energy sources.

This study builds on previous research by applying a variant of PSO, termed NvPSO, to optimize scheduling in AGVs, machine tools, and energy management. NvPSO enhances the swarm leader's position using a Gaussian distribution, broadening the search space for diverse solutions. To improve poorly performing solutions, Differential Evolution (DE) and mirroring techniques were incorporated, facilitating escape from local optima. Furthermore, particle position updates were refined using

rectified personal and global best signals, enhancing optimization precision. Chaotic inertia weights and dynamic switching probabilities balanced global and local search operations, while a spiral-based local exploitation strategy fine-tuned solutions, reinforcing local search efficiency.

2.3.5 Recent Advances in FMS and Flexible Job Shop Scheduling

2.3.5.1 Learning-Based and Reinforcement Learning Approaches

A dominant trend since 2025 is the application of deep reinforcement learning (DRL) and multi-agent reinforcement learning (MARL) to address dynamic and stochastic scheduling environments. Several studies model FJSSP as a Markov Decision Process (MDP), enabling schedulers to learn dispatching policies directly from evolving system states.

Shao et al. (2025) demonstrated that DRL-based schedulers outperform traditional dispatching rules under dynamic job arrivals and uncertain processing times. Extending this paradigm, Xu et al. (2025) proposed a MARL framework in which machines operate as autonomous agents coordinating local scheduling decisions, yielding improved robustness and throughput. Multi-objective extensions have also emerged; Li et al. (2025) developed collaborative RL approaches that simultaneously optimize makespan and energy consumption, highlighting the suitability of learning-based methods for adaptive and energy-aware smart factories.

Despite their promise, learning-based methods remain data-hungry and computationally intensive, particularly during training, which limits their standalone deployment in large-scale industrial settings.

2.3.5.2 Hybrid Metaheuristics and Graph-Based Optimization

Hybrid metaheuristics continue to dominate large-scale and constraint-rich FJSSP instances, particularly where solution quality and stability are critical. Recent work emphasizes problem-aware hybridization, combining population-based search with graph-based representations and local search.

Li and Zhang (2025) proposed a disjunctive graph–enhanced genetic algorithm integrated with targeted local search, significantly improving exploitation capability and mitigating premature convergence. Bio-inspired hybrids combining genetic algorithms, artificial bee colony methods, and problem-specific operators have also been reported to maintain population diversity while effectively handling fixture, tooling, and routing constraints.

Lv et al. (2025) introduced an enhanced Walrus Optimization Algorithm for FJSSP, achieving faster convergence and lower computational complexity, although solution quality degraded for very large instances—an observation consistent with findings in the present thesis. Overall, 2025 literature reinforces that hybrid metaheuristics remain indispensable, particularly when augmented with structural knowledge such as disjunctive graphs.

2.3.5.3 Energy-Aware and Sustainable Scheduling Models

Energy-aware scheduling has matured significantly since 2025, both methodologically and applicationally. Recent studies move beyond simplified idle-time energy models to incorporate machine power profiles, dynamic electricity tariffs, start–stop costs, and renewable energy availability.

Dunke et al. (2025) developed exact and approximate Pareto-based methods that jointly minimize makespan and energy cost under time-varying tariffs, demonstrating meaningful energy savings with limited productivity loss. Zhao et al. (2025) further extended energy modeling to include processing energy, idle energy, and AGV transportation energy, confirming that sustainability objectives must be embedded directly within the scheduling algorithm rather than applied post hoc.

Approximate multi-objective evolutionary algorithms remain preferred for large-scale instances, while exact and hybrid Pareto-generation approaches are shown feasible for moderate problem sizes.

2.3.5.4 Constraint Programming and Hybrid Exact–Metaheuristic Models

Renewed interest in constraint programming (CP) has been observed for FJSSP variants involving preventive maintenance, complex temporal constraints, and reliability considerations. Zhao and Wang (2025) proposed a CP formulation for maintenance-aware FJSSP and combined it with metaheuristic search to overcome scalability limitations.

Such CP-assisted metaheuristics provide strong bounds, certify solution quality for subproblems, and guide stochastic search—offering a promising pathway for integrating exact reasoning within heuristic frameworks.

2.3.5.5 Digital Twin and GenAI-Assisted Scheduling

Recent research increasingly emphasizes decision-support and deployment readiness rather than purely algorithmic performance. Digital twin frameworks have emerged as a critical enabler for closed-loop scheduling, allowing algorithms to interact continuously with real-time shop-floor data.

Kumar et al. (2025) demonstrated that embedding schedulers within CPS and digital twins significantly improves robustness to disturbances. In parallel, Chen et al. (2025) explored GenAI-assisted scheduling interfaces, enabling planners to generate, explain, and modify schedules interactively. In this context, GenAI enhances explainability and what-if analysis, while digital twins provide validation and execution fidelity—together forming a complementary ecosystem for Industry 4.0 deployment.

2.3.5.6 Human-Centric and Resilience-Oriented Scheduling

Recognizing the limitations of full automation, recent studies incorporate human availability, skill levels, fatigue, and operator preferences into multi-objective scheduling models. Rossi et al. (2025) showed that neglecting human factors often yields impractical schedules, particularly in semi-automated environments typical of MSMEs.

Additionally, resilience-oriented scheduling—focusing on robustness to machine failures, demand

fluctuations, and supply disruptions—has gained traction, often using multi-objective evolutionary and agent-based frameworks.

2.4 Inference from Literature Study

This section highlights research gaps, commonalities, and opportunities for further development in FMS scheduling, focusing on metaheuristics, objectives, and specific FMS-related challenges.

2.4.1 Metaheuristic Techniques

Solving complex scheduling problems using exact methods often requires extensive derivations and calculations, leading to high computational costs, especially as problem size increases. This becomes particularly challenging when dealing with NP-hard problems, where computation time grows exponentially. Metaheuristic approaches provide an effective alternative by offering near-optimal solutions with significantly reduced computation time.

In modern manufacturing, scheduling problems involve multiple objectives within tight timeframes, making the problem highly complex. The use of metaheuristics strikes a balance between solution quality and computational efficiency, making them well-suited for real-world optimization problems. Unlike exact methods, metaheuristic approaches do not impose rigid formulation rules for objective functions and constraints, allowing flexibility in problem-solving. Hybrid heuristics often outperform single-method approaches, particularly in multi-objective scenarios.

Among metaheuristics, Evolutionary Algorithms (EA) and Swarm Intelligence (SI) techniques dominate optimization research. SI-based methods are gaining popularity due to their simplicity and reduced computational complexity compared to EA. However, not all SI techniques effectively handle FMS scheduling, necessitating the development of optimized SI methods through parameter tuning or hybridization.

2.4.2 Multi-Objective Scheduling Techniques

Flexible manufacturing must adapt to customer demands, requiring efficient scheduling strategies to remain competitive. Traditionally, most studies have focused on minimizing makespan, but this single-objective approach does not always align with broader industry needs. With increased automation and parallel machine use, additional factors such as tardiness penalties and machine workload distribution must be considered.

A more comprehensive objective function integrating multiple scheduling goals is necessary. While some research categorizes problems with two objectives as multi-objective, a more rigorous analysis

often requires considering multiple competing objectives simultaneously. A flexible and adaptable optimization framework that accommodates varying production goals is essential for practical applications.

2.4.3 FMS Scheduling Challenges

A hierarchical approach is often used for Flexible Job Shop Scheduling (FJSP), addressing subproblems separately. However, an integrated approach that simultaneously optimizes machine assignment and routing could enhance efficiency by exploring a broader search space. This method increases search diversity but also adds complexity, necessitating effective search strategies to reduce computational demands.

Joint scheduling of machines and AGVs is crucial for minimizing production costs and makespan, yet few studies have explored simultaneous scheduling. Addressing this gap presents a significant opportunity for research, motivating the development of a robust hybrid algorithm tailored for FMS scheduling.

2.5 Research gaps

The analysis of existing literature on flexible manufacturing systems (FMS) has identified key research objectives by considering various scenarios. To adopt an effective meta-heuristic approach for the problem addressed in this study, the limitations and complexities of existing heuristic methodologies in FMS scheduling have been examined and outlined.

Research Gaps in PSO and GA Studies for Optimization

1. Limited Application of Hybrid PSO-GA Approaches
    - While some studies have explored hybrid PSO-GA models for scheduling and optimization, the full potential of parallel processing and hybridization techniques remains underutilized in large-scale and real-time industrial applications.
2. Lack of Adaptability in Existing PSO Variants
    - Existing PSO variants often struggle with balancing exploration and exploitation across highly dynamic optimization problems. Further research is needed to enhance adaptability using advanced learning-based mechanisms.
3. Insufficient Integration of Energy Optimization in Scheduling Models
    - Studies on scheduling optimization have primarily focused on makespan and efficiency.

However, energy-aware scheduling in flexible manufacturing remains underexplored, requiring new models that minimize energy consumption while maintaining high performance.

4. Limited Use of AI-Driven Predictive Models in Optimization

    - The integration of AI techniques (e.g., neural networks) into PSO and GA models for predictive scheduling and real-time adaptation is still in its early stages and needs further investigation.

5. Lack of Generalized Frameworks for AGV Scheduling

    - While MSSA and similar methods have optimized AGV scheduling, a generalized framework incorporating real-time constraints, stochastic factors, and adaptive learning is still needed.

6. Need for Multi-Objective Optimization with Practical Constraints

    - Current studies focus on theoretical improvements in optimization algorithms but lack practical implementation, especially in multi-objective scheduling that considers real-world constraints such as machine breakdowns, tool wear, and transportation delays.

7. Deficiencies in PSO-GA Hybrid Models for Real-Time Manufacturing Systems

    - The performance of PSO-GA hybrids in real-time scheduling scenarios, particularly in dynamic and uncertain environments, requires further testing and refinement.

8. Unexplored Potential of New Metaheuristic Techniques

    - Emerging algorithms like the Walrus Optimization Algorithm (WaOA) and Adaptive Evolutionary ABC (AEABC) have shown promise but remain underexplored in scheduling and optimization domains.

9. Gaps in Benchmark Testing for New Optimization Techniques

    - Many proposed optimization algorithms lack extensive benchmark testing against industry-standard problems, making it difficult to assess their real-world applicability.

10. Despite substantial progress, several gaps persist:

    - Limited online integration of hybrid metaheuristics with real-time CPS or RL-based reschedulers

- Incomplete energy models that insufficiently capture industrial tariffs, start–stop dynamics, and renewable integration
- Insufficient digital twin–based validation for live factory deployment
- Scalability challenges in distributed and multi-factory scheduling
- Underrepresentation of human-in-the-loop constraints in algorithm validation
- These gaps directly motivate extensions of the present thesis toward hybrid metaheuristic–RL integration, richer energy modeling, and CPS/digital twin embedding.

2.6 Summary

The literature review on flexible manufacturing systems (FMS) highlights various research objectives by analysing different scheduling methodologies. It has been observed that while heuristic approaches have been extensively used for FMS scheduling, they often face challenges in terms of efficiency, adaptability, and solution optimisation. Existing methods struggle with dynamic job allocation, resource constraints, sequencing flexibility, and real-time decision-making, all of which are critical in modern manufacturing environments. Additionally, energy consumption has become a significant concern, as traditional approaches fail to incorporate effective energy minimisation strategies.

To address these research gaps, this study employs advanced meta-heuristic techniques, including Novel Particle Swarm Optimisation (NvPSO), Genetic Algorithm (GA), Artificial Bee Colony (ABC) algorithm, and Whale Optimisation Method (WOM). These techniques enhance optimisation efficiency, adaptability, and scheduling performance while effectively reducing energy consumption. Furthermore, Pareto optimal solutions are utilised to ensure a balanced trade-off between multiple conflicting objectives, such as minimising makespan, improving resource utilisation, and reducing energy consumption. This comprehensive approach provides an efficient and sustainable scheduling strategy for FMS, overcoming the limitations of traditional heuristic methods.

CHAPTER 3
Meta Heuristic Techniques

3.1 Introduction

Meta Heuristic optimisation algorithms are a set of strategies used to discover the best feasible solution to complex problems involving several variables or intricate restrictions. Unlike traditional computing methods, which rely on hard mathematical models and precise rules, Meta-Heuristic approaches are motivated by biological processes and human reasoning. These techniques include artificial neural networks, fuzzy logic, evolutionary computation, and swarm intelligence. Meta Heuristic is especially useful when problems are poorly defined or the data is ambiguous or inadequate. These techniques have been used in a variety of industries, including engineering, medicine, finance, and natural language processing, demonstrating their versatility and adaptability to solving complicated problems in a changing world.

Meta Heuristic optimisation methods have been widely used in production scheduling, which entails defining work sequences, resource allocation, and job assignments in order to satisfy production goals while minimising costs and increasing efficiency. These algorithms are capable of exploring large solution spaces and identifying the best solution depending on certain criteria. An optimisation algorithm is a method for determining the optimal solution to a problem while sticking to predefined restrictions and objectives. These algorithms work by looking for alternative solutions that maximise or minimise the objective function while adhering to the constraints.

The working of **metaheuristic algorithms** typically involves a series of steps to guide the search for optimal or near-optimal solutions in complex problems. These steps may differ liable to specific metaheuristic used, but the general process follows a common pattern. Here's an overview of the typical steps:

3.1.1 Steps in the Working of Metaheuristic Optimization Algorithms:

1. **Problem Definition:**
    - Define the optimization problem clearly, including objectives, constraints, and decision variables. Identify the goal, such as minimizing or maximizing a certain function (fitness function).
2. **Initial Solution Generation (Initialization):**
    - Generate an initial population of candidate solutions (or a single solution, depending on the algorithm). These solutions can be created randomly or based on some initial conditions.
3. **Fitness Evaluation:**

    Evaluate the quality of each candidate solution using a fitness function (or objective function).

The fitness function tests how effectively each solution achieves the intended results or solves the problem.

4. **Selection (Exploration and Exploitation):**
    - Select solutions based on their fitness scores. The best-performing results are usually given a larger chance of being chosen for further refinement. Some algorithms also allow for random selection to maintain diversity.
5. **Variation (Exploration and Exploitation):**
    - Apply operators to the selected solutions to generate new candidate solutions. This step typically includes:
        - **Exploration**: Searching through new, unvisited areas of the solution space (e.g., using mutation or random perturbations).
        - **Exploitation**: Refining the current solutions by focusing on their improvement (e.g., using crossover or local search).
6. **Fitness Re-evaluation:**
    - Evaluate the newly generated candidate solutions by means of the fitness function again to regulate their quality.
7. **Choice of the Best Solution (Replacement):**
    - Based on the fitness scores, select the finest solutions for the next generation . Depending on algorithm, the selection could replace the worst solutions or use an elitist approach (preserving the best solutions across generations).
8. **Stopping Criterion:**
    - Check whether a predefined stopping condition is met. Common stopping criteria include:
        - A set number of iterations or generations.
        - A minimum threshold for improvement (e.g., no significant change in fitness over a set number of iterations).
        - The convergence of the population or solutions to a particular value.
9. **Final Solution (Termination):**
    - Once the stopping criterion is met, the algorithm terminates, and the best solution found during the search process is returned as the final solution to the optimization problem.

Additional Considerations:

- **Diversity Maintenance:** Many metaheuristic algorithms include mechanisms to maintain diversity in the population or solution set to avoid premature convergence to suboptimal solutions.

- **Stability of Exploration and Exploitation :** Metaheuristic algorithms aim to strike a balance between exploring and refining the best answers. This helps to prevent being caught in local optima.

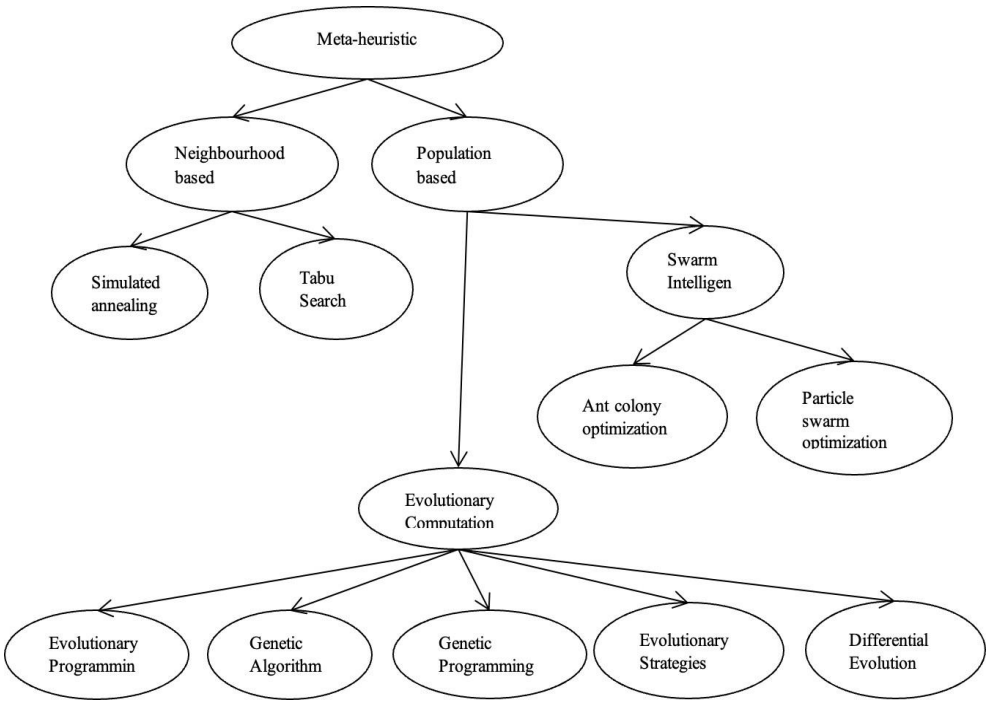


Figure 1: Classification of Meta Heuristic

3.1.2 **Metaheuristics inspired by Nature**

Nature-inspired metaheuristics use evolutionary processes to efficiently solve complicated problems. These natural mechanisms have been fine-tuned by evolutionary laws and have proven to be extremely successful in handling complicated natural challenges.

The core idea behind nature-inspired metaheuristics is to develop a set of candidate solutions and repeatedly refine them using a fitness function that measures how well each solution performs in meeting the optimization goal. The metaheuristic algorithms use this data to modify the search strategy and generate new candidate solutions.

Nature-inspired metaheuristic algorithms are frequently stochastic and iterative, producing a set of candidate solutions that are iteratively refined using a set of rules or heuristics derived from natural processes. These methods are frequently employed to solve optimization problems that are difficult or impossible to solve using conventional optimization algorithms.

Examples of natural events that have inspired metaheuristic algorithms are: Evolutionary processes include 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 metaheuristic algorithms listed below are developed and implemented to optimize scheduling of Flexible Manufacturing System and are discussed in greater detail in the following sections:

- **NG-AABCA – A Hybrid Genetic and Adaptive Artificial Bee Colony Algorithm with Pareto Optimality**

  NG-AABCA is a hybridization technique developed by integrating Artificial Bee Colony (ABC) and Genetic Algorithm (GA) within a Pareto-optimal multi-objective framework. Unlike conventional ABC, which suffers from weak exploitation and slow convergence, NG-AABCA incorporates cognitive (ε_1) and social (ε_2) learning components that introduce adaptive intelligence into the search process. The GA-based elitism mechanism ensures preservation of high-quality solutions, while random-restart hill climbing enhances local refinement and prevents premature convergence. Compared to classical GA, NG-AABCA maintains population diversity with significantly lower computational cost. As a result, the algorithm demonstrates superior robustness, faster convergence, and improved solution quality, achieving simultaneous reductions in makespan and total tardiness for flexible job shop scheduling problems, particularly in tardiness-sensitive industrial environments.

- **NvPSO – A Novel Variant of Particle Swarm Optimization**

  NvPSO is a novel variant of Particle Swarm Optimization specifically designed to overcome the limitations of conventional PSO in multi-objective and energy-aware manufacturing scheduling. The algorithm employs four primary enhancement strategies: (i) strengthening the swarm leader using a Gaussian distribution to improve global guidance, (ii) improving worst-performing particles through Differential Evolution (DE) and mirroring mechanisms to prevent stagnation, (iii) modifying the

position update equation using enhanced personal-best and global-best learning signals for better exploration–exploitation balance, and (iv) integrating a spiral-based local search strategy to intensify exploitation near promising regions. Unlike classical PSO and other swarm-based metaheuristics, NvPSO simultaneously optimizes productivity- and sustainability-oriented objectives, including makespan, tardiness, and energy consumption. The algorithm demonstrates robust performance across diverse job sets and consistently outperforms traditional PSO, AIS, and MGTA, while achieving notable energy savings and faster convergence compared to WaOA, making it well suited for sustainable manufacturing applications.

- **Dy-PSO – Dynamic Particle Swarm Optimization**

  Dy-PSO is an advanced dynamic variant of Particle Swarm Optimization developed to address premature convergence, swarm stagnation, and scalability issues inherent in classical PSO. In Dy-PSO, the population is divided into multiple interacting sub-swarms that periodically exchange information through a structured process of splitting and regrouping. After a predefined regrouping period (R), all sub-swarms are merged and re-divided, enabling effective knowledge sharing across the entire population. This dynamic restructuring allows the algorithm to adapt continuously to complex and changing search landscapes, ensuring a strong balance between exploration and exploitation. Unlike conventional PSO, Dy-PSO exhibits superior scalability and robustness for high-dimensional flexible manufacturing system scheduling, including simultaneous coordination of machines, tools, and AGVs. The integration of learning-based guidance further enhances convergence speed and solution stability, resulting in consistently lower makespan and improved performance for large-scale industrial scheduling problems.

- **Overall Superiority of the Proposed Optimization Framework**

  Collectively, the proposed NG-AABCA, NvPSO, and Dy-PSO algorithms represent a significant advancement over conventional metaheuristics such as GA, PSO, ABC, SA, and TS. Their superiority lies in the systematic integration of adaptivity, hybridization, learning mechanisms, and multi-objective optimization, enabling efficient handling of simultaneous scheduling, energy-aware decision-making, and large-scale industrial problem instances. The demonstrated scalability, robustness, and validation through realistic Indian manufacturing case studies further highlight the industrial relevance and practical applicability of the proposed framework.

3.2 NG-AABC- Hybridisation of ABC and GA with Pareto optimality

This study proposes an integrated method that combines the Novel Genetic Algorithm (GA) with the Adaptive Artificial Bee Colony Algorithm (AABC), referred to as NG-AABC. Cognitive learning (ε1) from the ABC is incorporated into the GA, allowing each chromosome to remember its past performance. This favors individuals with a history of better solutions during selection. Additionally, social learning (ε2) from the ABC is simulated by enabling genetic material exchange among specific individuals in the GA population, emulating the sharing of valuable information within the ABC. A balance between exploration and exploitation is crucial and is achieved through the social component (to exploit known solutions) and the cognitive aspect (to explore new possibilities). Furthermore, a novel GA is introduced to improve solution quality, preserve elite chromosomes, and enhance diversity by balancing parameters. Dynamic adjustment of crossover, mutation, and selection probabilities is applied to maintain this balance. The effective tuning of parameters ensures the smooth interaction of both algorithms. Ultimately, this hybridization aims to harness ABC's memory and information-sharing strengths while leveraging GA's genetic optimization capabilities, thus enhancing the ability to tackle complex simultaneous scheduling problems.

3.2.1 Genetic Algorithm (GA) Overview:

Genetic Algorithms (GAs), inspired by the principles of genetics and natural selection, are heuristic search methods that originated in the 1970s. Their core concept is to identify the optimal individuals whose chromosomes lie within the search space. By combining elitism and local search, GAs can adapt more effectively and exploit promising regions identified by elite individuals, while refining solutions via local optimization. This hybrid approach helps the GA escape local minima and converge towards high-quality solutions. When combined with the Adaptive-ABC Algorithm, this hybridization strengthens exploration and exploitation, improves parameter tuning, and avoids local minima, ultimately leading to better solutions.

3.2.1.1 Chromosome Representation:

Chromosomes represent job permutations, with a focus on operation sequences and machine assignments. Each chromosome encodes a collection of numbers corresponding to jobs, operations, and machine sequences. For instance, a chromosome may represent job assignments like O21 on machine M1, with values corresponding to job operations and machine indices. The GA uses crossover and mutation operations to explore and optimize scheduling solutions, ensuring adherence to job precedence and machine assignment constraints.

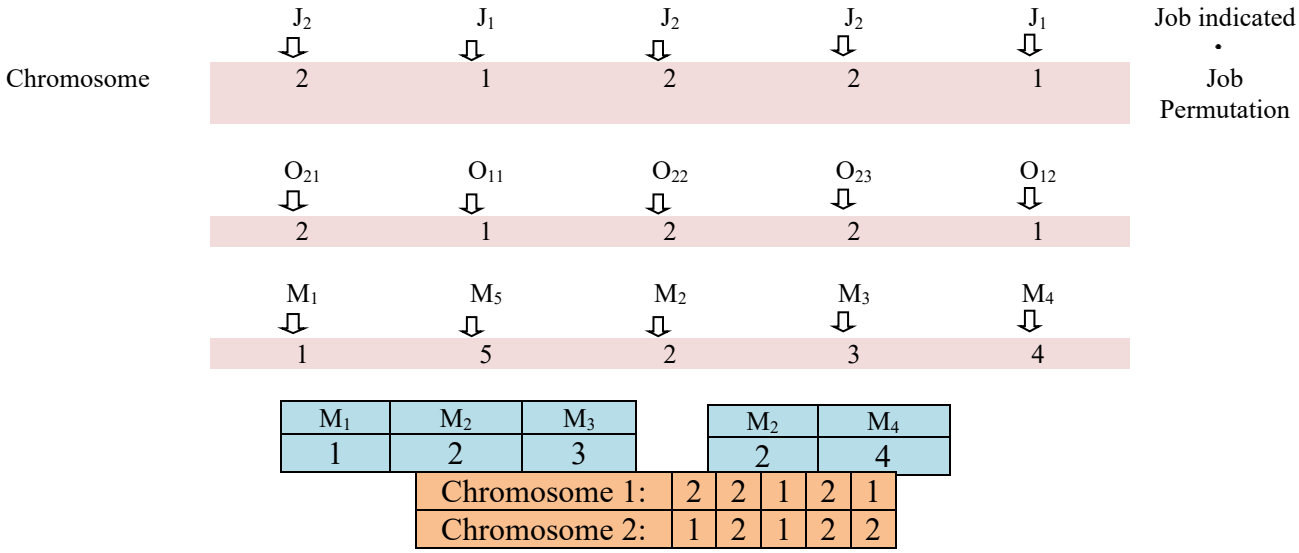


Figure 2: Chromosome Structure

3.2.1.2 Elitism-based Genetic Algorithm:

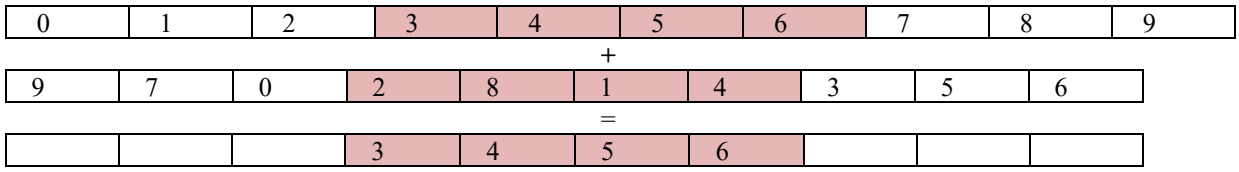
Elitism involves transferring the best-performing chromosomes from one generation to the next, thereby preventing the loss of top solutions during crossover and mutation. By incorporating elite chromosomes, the algorithm avoids premature convergence and helps maintain population quality. Elite chromosomes are selected based on their fitness and advanced to the next generation without modification, thus accelerating the algorithm's performance.

$$\text{JCi} = \sum_{i=1}^{n} OT\ \text{i}j \qquad (1)$$

$$\text{Mean Tardiness} = \frac{1}{n}\sum_{i}^{n} Ti, \qquad (2)$$

3.2.1.3 Crossover and Mutation Operations:

Crossover and mutation operations are essential for generating new solutions. In the crossover step, two parent chromosomes are combined to produce offspring. A single-point crossover ensures that the resulting offspring is a valid solution. Mutation introduces random changes to offspring chromosomes, preserving diversity in the population. A swap mutation, for example, may modify two values on a chromosome, allowing the algorithm to explore the solution space and avoid becoming stuck in local optima.



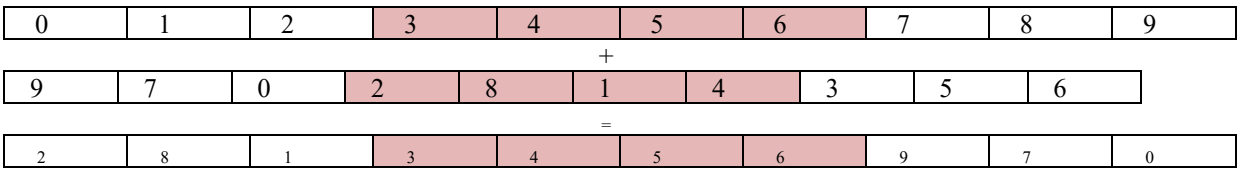


Figure 3. Crossover Operation

If we apply swap mutation by interchanging the values at positions 2 and 6, the resulting chromosome becomes:

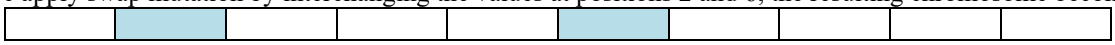


Figure 4. Mutation Operation

3.2.1.4 Random-Restart Hill Climbing (RRHC):

The RRHC method enhances the GA by introducing a local search technique. After creating an initial population of solutions, the GA employs genetic operators to generate additional solutions.The best solution from this set undergoes further refinement through RRHC, which restarts after a set number of iterations without improvement. This cyclic process helps escape local optima and ensures continuous improvement until the termination criterion is met.

Initialization:
- receive the initial solution s, $iterations\_HC$ and $n\_restart$;
- evaluate $f(s)$;
-set $i\_restart = 0$;
for $i = 1$ to $iterations\_HC$
- if $i\_restart > n\_restart$ restart the initial solution s stochastically from s';
$s \leftarrow$ new initial solution;
$i\_restart = 0$;
- end;
$s' \leftarrow$ one new neighbors generated from s;
- evaluate $f(s')$;
- if $f(s') < f(s)$, set $s = s'$, else $i\_restart++$ **end**
for
end

Figure 5:Random-Restart Hill Climbing Algorithm

3.1.2.1 ABC Algorithm

The Artificial Bee Colony (ABC) algorithm is based on the foraging behaviour of bee colonies. It separates the colony into three sorts of bees: employed bees, observation bees, and scout bees. The employed bees look for food sources (solutions), the observer bees evaluate food sources, and the scout bees investigate novel sources at random. The fitness of a food supply symbolises its utility, while the ABC algorithm seeks optimal solutions by mimicking bee colony behaviour.

Adaptive-ABC Algorithm:

The Adaptive-ABC algorithm improves upon the basic ABC by introducing dynamic learning factors (ε1 and ε2). These factors adapt throughout the algorithm's search process, balancing the exploration and exploitation capabilities. The learning factors change depending on the no of iterations, permitting

the algorithm to explore more broadly at the start and focus on refining solutions as the search progresses.

3.1.2.2 Onlooker and Scout Bee Phases:

In the observer bee phase, bees assess and choose food sources depending on their fitness. A greater fitness score enhances the chances of selection. If a food source is not improved after a certain number of repetitions, the bee transforms into a scout and hunts for new food sources at random.

NG-AABC Algorithm Integration:

The NG-AABC method blends the strengths of both the GA and the Adaptive-ABC algorithm. Initially, the AABC population consists of the best solution from the GA, alongside random solutions. The AABC proceeds by initializing the population and exploring the solution space using ε1 and ε2 for global exploration. The algorithm continues until the termination criterion is met, and the optimal solution is returned.

Hybrid NG-AABC Algorithm:

Step 1: Initialization

1. Initialize parameters:
    - Set the parameters for both the Genetic Algorithm (GA) and Adaptive ABC (AABC) algorithms:
        - Population size (N)
        - Maximum generations (Gmax)
        - Crossover, mutation probabilities (pc, pm)
        - Selection probabilities for GA
        - Number of food sources (SN) for ABC
        - Exploration and exploitation parameters (ε1, ε2)
        - Termination criterion (e.g., maximum generations or fitness threshold)
2. Initialize the population:
    - Generate an initial population for GA by randomly selecting chromosomes representing job permutations and machine assignments.
    - For ABC, initialize the food sources with random solutions in the search space.
3. Evaluate the fitness of the initial population for both GA and ABC using the fitness function (e.g., makespan, tardiness).

Step 2: GA Operations

4. Selection:
    - Choose individuals based on fitness through a tournament or roulette-wheel selection.

This means that persons with higher fitness levels have a better chance of being chosen.

- Implement elitism, where the top p% of the population (based on fitness) are passed down unchanged to the following generation.

5. Crossover:
    - Perform crossover (single-point or two-point) between pairs of selected individuals to create offspring.
    - Ensure that offspring respect job precedence and machine assignments while performing the crossover.
6. Mutation:
    - Perform mutation (swap mutation) to maintain genetic diversity. This involves randomly swapping values in the chromosome to introduce new combinations.
    - The mutation rate (pm) controls the probability of mutation.
7. Elitism Update:
    - After performing crossover and mutation, select the best individuals from the current generation and carry them forward to the next generation.
    - This helps prevent premature convergence and ensures the retention of high-quality solutions.
8. Fitness Evaluation:
    - Evaluate the new generation's fitness population after crossover and mutation.

Step 3: Adaptive-ABC Operations

9. Employed Bee Phase: Use the algorithm to create a new food source for each employed bee near the present one.

    $$hnew(j)= \varepsilon 1hij + \varepsilon 1c1 (hij- hkj) X r + \varepsilon 2 c2 (hij- hkj) X r \quad (3)$$

    hi(j)h_i(j) is the current food source (solution),
    - hk(j)h_k(j) is another randomly chosen food source,
    - r1,r2r_1, r_2 are random values between [-1, 1],
    - ε1\varepsilon_1 and ε2\varepsilon_2 are dynamic learning factors that vary over iterations.
10. Fitness Evaluation:
    - Evaluate the fitness of the new food source.
    - If the new solution is better or equal to the previous one, replace the old food source with the new one and reset its iteration counter.
    - If not, keep the old solution and increase the iteration counter.

11. Onlooker Bee Phase:

Onlooker bees evaluate food sources and choose the optimum solution based on fitness likelihood.

$$Pi=1/fi \,/\sum iSN=1\ 1\ /fi \qquad (4)$$

where fif_i is fitness of ii-th food source.

- After selecting a food source, the observer bee produces a new solution similar to the employed bee's behaviour.

12. Fitness Evaluation:

- Evaluate the fitness of the newly selected solution and update the population if the new solution is better.

13. Scout Bee Phase:

- If a food source has not improved for a set number of iterations (limit), the employed bee becomes a scout. The scout bee explores a random new food source (solution) in the search space:

$$hij=Lj+(Uj- Lj)X\ r \qquad (5)$$

$$j = 1, 2, 3,4,. . ., n$$

where r is a random number with the lower and upper range of 0 and 1.

14. Fitness Evaluation:

- Evaluate the fitness of the newly generated food source.
- Consider replacing an old food source with a better one.

Step 4: Hybridization and Termination

15. Hybridization:

- Integrate the best individuals from both GA and AABC populations:
  - The elite chromosomes from GA are mixed with the top solutions found by ABC.
  - The best solutions from both algorithms are retained for the next iteration.

16. Stopping Criteria:

- If the termination criterion is met (e.g., maximum generations or convergence threshold), stop the algorithm.
- Otherwise, return to Step 2 (GA operations) and Step 3 (ABC operations) for the next iteration.

Step 5: Output the Best Solution

17. Final Solution:

- Once the termination requirement is met, output the hybrid algorithm's optimal solution.

- This is the optimal or near-optimal solution for the scheduling problem.

3.3 Novel variant of Particle Swarm Optimization (NvPSO)

In this algorithm, a novel variant of Particle Swarm Optimization (NvPSO) is introduced to overcome key challenges in the traditional PSO model, specifically early convergence and inadequate local exploration of Suboptimal solutions. This variant employs four primary strategies: enhancing the swarm leader using Gaussian distribution, improving the worst-performing solutions through Differential Evolution (DE) and mirroring techniques, modifying the position update mechanism with improved personal and global best signals, and integrating a spiral-based approach for local exploitation. Together, these strategies boost population diversity and search efficiency, significantly growing chances of achieving global optimality compared to the standard PSO algorithm. Notably, this variant utilizes refined forms of personal and global best signals, a chaotic inertia weight, and dynamic switching probabilities to balance global and local search operations. Additionally, it incorporates Gaussian distribution for enhancing swarm leaders and mutation strategies to strengthen weaker particles. These innovations work together to reduce the likelihood of becoming stuck in local optima and improve overall optimisation performance.

3.1.3.2 A Method to Augment the Swarm Leader:-

In the realm of scheduling, the removal of ideal time and the inclusion of conflicting attributes can significantly impact classification performance. To cater this, we propose a mechanism to enhance the swarm leader, gbest, using skewed Gaussian distributions. This approach aims to endow gbest with enhanced discriminative capabilities, helping it navigate through local optima challenges(Jordehi, 2015). The mutation of gbest is performed successively using three Gaussian distributions with varying skewness settings, as expressed in Equation (6). Specifically, a right-skewed distribution is employed to potentially eliminate longer makespan, while a left-skewed distribution is more likely to capture distinctive makespan. Additionally, a standard Gaussian distribution, without skewness, is utilized for local exploitation of gbest while maintaining neutrality.(Yang, 2014)

$$gbestb' = gbestb + \beta \times Gaussian(g) \times (Ub - Lb) \quad (6)$$

In this context, gbestb represents improved global best results. The " β " indicates the pace size, which was set at 0.2 based on references from relevant literature (Yang, 2014). The parameter g signifies the skewness of the Gaussian distribution, with values of 2, 2, and 0 assigned for left-skewed, right-skewed, and non-skewed distributions, as determined through wide experimentation. Furthermore, Ub & Lb refer to the higher and lower limits of dth dimension. The output produced by the Gaussian distribution is used to substitute gbest if it demonstrates a better fit as a solution.

3.1.3.3 Improving the Weakest Solution in the Swarm through Mutation-Based Augmentation.

We improve the weaker elements in the swarm using a two-step approach that includes mirroring mutation applied to swarm leader & Differential Evolution for local elite solutions. Initially, we apply a gbest-based local transformation approach, as outlined in Equation (7), which generates a new particle by utilizing mirroring effects and inverting the sign of gbest. This process is influenced by a mutation probability, rmu, across each dimension. This process replicates random activation or deactivation of

specific features grounded on the existing optimal feature subset indicated by gbest. Essentially, the gbest-based neighbourhood mutation strategy maintains stability amid retaining valuable data from the gbest solution and presenting stochastic variations to stimulate novel progress in the resulting solution. Similar reflecting methods have been effectively utilized in previous research (Yang, 2014) to boost population diversity.

$$w_d^{new} = \begin{cases} -\text{gbestb if rand } \geq \mu_m \\ \text{gbestb otherwise} \end{cases} \quad (7)$$

The mutation probability, represented as μm and set to 0.8 based on experimentation along with insights from related studies (Jordehi, 2015), regulates the generation of a new offspring. Specifically, if the arbitrary value is ≥ μm , the following generation either inherits a reflective value from the gbest solution in the dth dimension or adopts the value from the gbest solution for that dimension. The mechanism aims to create a new generation that may substitute underperforming particles in the swarm if it shows improved performance.

Additionally, a Differential Evolution approach is utilised to enhance the 2nd and 3rd weakest entities within the swarm. This technique generates new particles through DE's mutation & crossover processes, utilizing three randomly chosen pbest solutions from the entire pool of pbest entities,in Equations (8) & (9). The differential weight, F, in Equation (16) is created using a Sinusoidal chaotic map which introduces varied disturbances for the donor vector, y_d^{donor}, across each dimension. The crossover parameter CO is determined by a Logistic chaotic map, adding further randomness to the crossover process and promoting greater feature interactions globally. If the arbitrarily generated value exceeds cr, the corresponding dimension in the new solution is derived from the pbest solution; if not, it is taken from the afresh generated donor solution. By incorporating a variety of pbest solutions into the exploration operations, this Differential Evolution-based mutation strategy significantly enhances population range particularly when the pbest solutions of the entitities exhibit considerable variation during the initial phases of the search.

$$y_d^{donor} = pbest_d^1 + f\ X(pbest_d^2 - pbest_d^3) \quad (8)$$

$$w_d^{new} = \begin{cases} y_d^{donor} \text{ if rand } \geq C_o \\ pbest_d^1 \text{ otherwise} \end{cases} \quad (9)$$

In the dth dimension, $pbest_d^1$, $pbest_d^2$, and $pbest_d^3$ signify 3 arbitrarily nominated pbest solutions. The donor & new solutions in dth dimension are referred to as y_d^{donor} and w_d^{new} respectively. Additionally, the symbols f and C_o correspond to the differential weight and crossover factor.

The afresh-created solution is acknowledged immediately if it exhibits superior fitness. However, if the mutated solution is less fit, its acceptance is established by an annealing plan described in Equation (10). In this equation, T signifies the temperature that governs the annealing process, while f designates the fitness difference between the mutated and original solutions. A random value within the range of [0, 1] is also incorporated as a constant. A linear cooling schedule is applied to gradually decrease the

temperature, where T is multiplied by 0.9, as noted by (Jordehi, 2015).Both mutation processes, based on DE and gbest mirroring, function concurrently to improve the weaker particles.

$$p = exp\left(\frac{\Delta f}{T}\right) > \delta \quad (10)$$

3.1.3.4 Approach for Enhanced Diversity in PSO Evolution

To tackle stagnation issues in the unique PSO model, we introduced 2 new exploration mechanisms: a modified PSO search approach & a strengthened spiral exploitation approach. These strategies aim to enhance both diversification and intensification. Dynamic switching prospect schedule is also implemented to effectively balance these strategies, maximising both search operations' advantages.

First, we improve the position-apprising approach of the original PSO by incorporating enhanced (pbest) and (gbest) values, along with the Logistic chaotic map, to boost search range. Using (11) equation , we refine gbest by averaging positions of the current gbest solution and a neighbouring Better pbest solution, which we refer to as pbest. This pbest is selected based on its dissimilarity to gbest, measured by counting the no of distinct units in binary representations. Essentially, pbest is chosen as the solution that shares the fewest features with gbest. Equation (20) improves the local best experience by averaging particle's pbest with another arbitrarily selected superior pbest solution, labelled as pbestR. Equation (13) then governs the position updating process.

$$gbest_d^M = (gbestd + pbest_d^D)/2 \quad (11)$$

$$pbest_d^M = (pbestid + pbest_d^R)/2 \quad (12)$$

$$V_{id}^{t+1} = \sigma \text{ x } V_{id}^t + \text{c1 x r1 x}(pbest_d^M - x_{id}^t) + \text{c2 x r2 x } (gbest_d^M - x_{id}^t) \quad (13)$$

3.1.3.5 Enhanced Structure for Spiralled Exploitation

We introduce an enhanced method for spiral exploitation aimed at addressing the refining of limitations of the original PSO algorithm, particularly in vicinity of optimum areas. This approach employs a logarithmic spiral search, inspired by the MFO algorithm, to refine the positions of swarm particles during the final iterations. By constructing a hyper-ellipse search space around the global best (gbest) using the spiral function ((14) and (15)), the local search around near-optimal solutions is greatly enhanced. In this framework, D represents the gap between gbest and particle i in the dth dimension, b is a constant that defines outline of the logarithmic spiral. Furthermore, a dynamic switching probability schedule (Equation (16)) is introduced to balance global exploration and local exploitation in this model. Switching probability, pswitch, is calculated based on the current iteration (iter) relative to the maximum iterations (Max_iter). During each iteration, if pswitch is greater than a arbitrarily generated value between [0, 1] the modified PSO operation takes place; otherwise, the spiral search is executed. This dynamic approach ensures sufficient global exploration in the early stages to locate promising regions, while also enabling thorough exploitation around near-optimal solutions as process converges in the final iterations.

$$x_{id}^{t+1} = \text{D x exp(b x l) x cos}(2\pi \text{ l}) + \text{gbestd} \quad (14)$$

$$\text{D} = \text{gbestd} - x_{id}^t \quad (15)$$

pswitch = 1- (iter/Max_iter)2 (16)

3.1.4.1 The Walrus Optimization Algorithm (WaOA)

The walrus, a marine mammal found in Arctic and subarctic waters, exhibits distinct behaviours including foraging guided by individuals with the longest tusks, migration to rocky beaches during warmer weather, and strategic responses to predators like polar bears and killer whales(Trojovský & Dehghani, 2023). These behaviours serve as a stimulus for the development of the Walrus Optimization Algorithm (WaOA). WaOA utilizes principles such as guided foraging, migration towards diverse destinations, and adaptive responses to threats, aiming to enhance global search and prevent premature convergence in optimization processes.

The Walrus Optimisation Algorithm (WaOA) is a population-based optimisation method that represents individuals as walruses, with each walrus representing a potential solution to an optimisation issue. In WaOA, each walrus' position in the exploration space correlates to the values of the problem variables, effectively transforming each walrus into a vector. These walruses are organised in a matrix known as the population matrix. To begin the WaOA implementation, this population matrix is produced at random to initiate the process.

$$Y = \begin{bmatrix} Y1 \\ \vdots \\ Yn \end{bmatrix} = \begin{bmatrix} y11 & \cdots & \\ \vdots & \ddots & \vdots \\ & \cdots & \end{bmatrix}$$

In the matrix Y, each row Yi signifies a walrus, serving as a potential solution to the problem, with Yi,j representing the j th decision variable's value recommended by that walrus. The objective function, calculated based on these values, yields an objective function vector (F), where (Fi) denotes the value obtained from i-th walrus.

$$Y = \begin{bmatrix} F1 \\ \vdots \\ Fn \end{bmatrix} = \begin{bmatrix} f(Y1) \\ \vdots \\ f(YN) \end{bmatrix}$$

These objective function values are pivotal for assessing solution quality. The walrus corresponding to the best value is termed best member, while one with poorest value is the worst member. By each iteration, as objective function values are updated, so too are the identities of the best and worst members.

$$y_{i,j}^{p1} = y_i + rand_{i,j}\left(SW_y - I_{ij} \cdot y_{i,j}\right) \quad (17)$$

$$y_i = \begin{cases} y_i^{p_1} F_i^{p_1} < F_i \\ y_{i'}\ else \end{cases} \quad (18)$$

The WaOA is mathematically modelled in three phases, mirroring the natural behaviours of walruses. The first phase, akin to the feeding strategy of walruses, focuses on exploration. Walruses, with their varied diet, predominantly target benthic bivalve mollusks, and clams, by foraging around sea level guided by the strongest member with the longest tusks. Similarly, in WaOA, the solution with the finest objective function value serves as strongest walrus, guiding others to discover different areas of the

search space. The algorithm updates walruses' positions based on this guidance, aiming to enhance global exploration. This process is mathematically expressed through equations (19) and (20), where a new position is created and replaces the previous one if it outcomes in an enhancement of the objective function value.

$$y_{i,j}^{p2} = \begin{cases} y_{i,j} + rand_{i,j}\left(x_{k,j} - I_{ij} \cdot y_{i,j}\right), F_k < F_i \\ y_{i,j} + rand_{i,j}\left(y_{i,j} - y_{k,j}\right), else \end{cases} \quad (19)$$

$$y_i = \begin{cases} y_i^{p_1} F_i^{p_1} < F_i \\ y_{i'}\ else \end{cases} \quad (20)$$

In the first phase, the Walrus Optimization Algorithm (WaOA) updates the positions of walruses to enhance exploration. New positions y_i^{p1} for each walrus is generated based on Equation (19), with $y_{i,j}^{p1}$ denoting the j -th dimension of the position. These positions are determined by random numbers $rand_{i,j}$ and an integer I_{ij}, randomly selected between 1 and 2, which influences the extent of position changes. The finest candidate solution, referred as strongest walrus (SW), directs exploration process. This phase focuses on enhancing global search by promoting exploration of various regions within the problem-solving space. In the second phase, reflecting the migration behaviour of walruses, the algorithm encourages them to investigate different areas of the exploration space. This migration procedure is mathematically characterized by Equations (21) and (22). Each walrus moves to a new position, randomly chosen from another walrus's location in the search space. If the new position leads to an improved objective function value, it substitutes the walrus's previous position. This migration process helps uncover more favorable regions in the search space, enhancing the algorithm's overall search efficiency. In Phase 3 of the WaOA (Walrus Optimization Algorithm), the behavior of walruses escaping and fighting predators is simulated to enhance local search around candidate solutions. This is achieved by adjusting the position of each walrus within a dynamic, walrus-centered neighborhood. Initially, the algorithm uses a broader radius for global exploration, which decreases focus on local exploitation as the iterations progress. New positions are randomly generated within this neighbourhood, and if they improve the objective function, they replace the previous ones, thereby refining the search for an optimal solution.

$$y_{i,j}^{p3} = y_{i,j} + \left(lb_{local,j}^{t}\right) + \left(\left(ub_{local,j}^{t}\right) + - \text{rand.}\left(lb_{local,j}^{t}\right)\right), \quad (21)$$

$$\text{Local bounds}: \begin{cases} lb_{local,j}^{t} = \frac{lb}{t} j \\ ub_{local,j}^{t} = \frac{lb}{t} j \end{cases} \quad (22)$$

$$y_i = \begin{cases} y_i^{p_3} F_i^{p_3} < F_i \\ y_{i'}\ else \end{cases} \quad (23)$$

In this context, $y_i^{p_3}$ represents the updated generated location for i th walrus during the third phase. The component $y_{i,j}^{p3}$ denotes the jth dimension of this position, while $F_i^{p_3}$ is corresponding objective function

. The variable t refers to current iteration, and lbj and ubj are the lower and upper bounds. Additionally, $lb_{local,j}^{t}$and $ub_{local,j}^{t}$ are the local lower and upper bounds for the jth variable, which are utilized to perform a local search within the neighborhood.

3.4 Particle Swarm Optimization (PSO)

Particle Swarm Optimisation (PSO) is a metaheuristic algorithm commonly employed to solve optimisation problems. It operates on a population of particles, each representing a potential solution inside a D-dimensional search space. The population consists of N particles, each characterized by a position and a velocity. The iterative process of PSO involves updating these attributes to find an optimal solution.

Key Features of PSO:

- Personal Best (pbest): Each particle retains its best-known position based on its own performance.
- Global Best (gbest) : The overall best position achieved across the population.
- Vk (t+1)= w* Vk(t) +c1*r1* (pbestk(t)- Xk(t)+c2*r2(gbest(t)-Xk(t)) (24
  Xk(t+1)=Xk(T)+Vi(t+1) (25)
- Where V_k is the velocity vector, X_t is the position vector,c_1 & c_2 acceleration coefficients and r1 & r2 cognitive acceleration and social acceleration.
- PSO uses pbest and gbest to guide particles toward optimal solutions. However, being a metaheuristic algorithm, PSO often requires numerous iterations, resulting in prolonged computation times compared to heuristic and hyper-heuristic approaches.
- Enhancing Computational Efficiency in PSO
- To address the computational limitations of PSO and improve efficiency, the following measures have been proposed:

3.4.1. Adaptive Parameter Adjustment

Dynamic-PSO incorporates adaptive strategies to adjust parameters during iterations, which include:

- Updating Stagnant Populations: This strategy prevents particles from stagnating and converging prematurely to local optima, thereby reducing redundant iterations.
- Spatial Excursion Techniques: These techniques promote effective exploration of the solution space, enabling a broader search for optimal solutions.

2. Simplified Encoding and Decoding Scheme

A streamlined encoding and decoding mechanism is utilized, particularly for complex problems like scheduling. This simplification reduces computational overhead and accelerates the solution process. By integrating these enhancements, PSO's computational efficiency is significantly improved, making it more practical for solving real-world optimization problems while maintaining solution quality.

3.4.2 Dynamic Multi-Swarm Particle Swarm Optimization

Dy-PSO, an abbreviation for Dynamic Multi-Swarm Particle Swarm Optimization, represents an effective adaptation of PSO renowned for its capability to address intricate optimisation challenges (Liang & Suganthan, 2005). The core concept driving Dy-PSO involves the utilisation of multiple collaborating swarms, as opposed to a sole population model. This strategy increases the odds of discovering the global optimum during the optimisation process.

In Dy-PSO, the population is separated into several sub-swarms, and these sub-swarms often exchange information with each other through splitting and regrouping. The regrouping process occurs after reaching a preset regrouping period (R). During regrouping, all sub-swarms are merged, and then they are split again into new sub-swarms. The purpose of this frequent splitting and regrouping is to enable effective information exchange among all the sub-swarms, allowing them to benefit from each other's exploration and exploitation capabilities.

Therefore, the regrouping period (R) is a crucial constraint in Dy-PSO as it determines rate at which information exchange occurs among the sub-swarms. A shorter regrouping period would lead to more frequent information exchange and potentially faster convergence, but it may also increase computational overhead. On the other hand, a longer regrouping period reduces the frequency of information exchange, allowing more independent exploration but potentially slowing down convergence. Selecting an appropriate value for the regrouping period is essential to balance exploration and exploitation in Dy-PSO and achieve effective optimization performance. The following formula (26) is used instead of the above (25) which finds the local best instead of sub-swarms instead of a single population.

$$Vk\ (t+1)= w^* Vk(t) +c1^*r1^* (pbestk(t)- Xk(t)+c2^*r2(lbest(t)-Xk(t)) \quad (26)$$

3.4.3 Methodology for population state monitoring

This technique improves on the outdated Dy-PSO algorithm, in which sub-swarms update individually until a certain regrouping period is achieved. Insufficient information exchange among sub-swarms can cause them to become caught in local optima. To solve this, a study is proposed that uses the best particle from each sub-swarm to evaluate its evolutionary state and then applies different information sharing mechanisms to improve the chances of finding the global optimum in the optimisation process.

This approach aims to enhance information exchange and improve optimization efficiency. Each sub-swarm functions independently, and the variation in best from one generation to the next ($\Delta lbestk$) serves as an indicator of the sub-swarm's evolutionary state as expressed in Eq. (27). In this equation, t denotes the number of iterations, while k refers to the sub-swarm's identifier. When dealing with minimization problems, two situations can arise.

$$\Delta lbestk(t)=f\ [lbestk(t)] -f\ [lbestk(t-1)]\leq 0\ k=1,2,\ldots..,M\ldots\ldots\ldots\ldots.. \quad (27)$$

when

$\Delta lbestk = 0$ or

$\Delta lbestk < 0$

In the first case, if the best solution of the kth sub-swarm remains unchanged for t generations, it indicates stagnation. To maintain a rich variety within the system and counteract diminished diversity, a stagnation factor denoted as η increases by 1 whenever the best solution doesn't undergo an update. As soon as η surpasses a predefined threshold labelled as S, a vitality particle is introduced to amplify diversity. The adaptive tuning of parameter S enhances the algorithm's effectiveness by regulating how often information is exchanged between inactive subgroups.

Grounded on the results of experiment, the upper and lower limits are set to Smax = 20 and Smin = 10, respectively in equation (28).

$$S = Smax - (Smax - Smin) * fes / Max\ Fes \quad (28)$$

In alternate scenarios, the best solution within the kth sub-swarm has undergone enhancement during the t generation. In such instances, this specific sub-swarm has identified a favourable position, leading to a reset of the stagnation factor to zero.

3.4.4 Methodology for updating a stagnant population

Vitality particles are key in the stagnant population update mechanism, aiding stagnant sub-swarms in escaping local optima. Constructing high-quality vitality particles is essential for enhancing population diversity. The construction process involves utilizing outstanding past information from all particles in the entire population.

$$P = (pbest1 , pbest2 , pbest3 \ldots\ldots\ldots\ldots .. pbestk) \quad (29)$$

$$Qk,worst = \max \{ f(xk,1) , f(xk,2) \ldots\ldots f(xk,n) \} \quad (30)$$

$$Nk = Nk / Gk,worst \quad (31)$$

Where P stands for elite pool and is refreshed after every group, Gk,worst is the worst particle in the sub-group and n is the population size.

During every generation, the individual best solutions (pbest) for all particles are preserved within the elite pool (E). Within this pool, the poorest-performing particle in the kth inactive sub-swarm is pinpointed and excluded. Subsequently, two distinct particles are arbitrarily chosen from the elite pool . By comparing their performance, a dynamic particle (e) is created to encompass both accuracy and diversity insights. This sequence guarantees that the vitality particles encompass a blend of precision and a variety of details.

By leveraging historical information and selecting the most favourable particles in each dimension, the vitality particles contribute to enhancing diversity within the population.

$$ek,d = \begin{cases} Er_{1,d} & f(Er1) \leq f(Er2) \\ E_{r_2,d} & , otherwise \end{cases} \quad (32)$$

ek,d = rand * (Ud – Ld) + Ld (33)

Vitality particles are constructed using random integers $r1$ and $r2$ within the range [1, N]. The construction process involves comparing performance and applying a mutation operation. A random number determines whether to use Eq. (26) or Eq. (27) for each dimension. The parameter ρ controls the level of randomness, and its adaptive adjustment depends on the population's evolution state.

According to historical data, ρmax = 0.17 & ρmin = 0.12 is the best values.

ρ = ρmax – (ρmax – ρmin) * fes/ Max Fes

Vitality particles are constructed by comparing the fitness of ek and Gk,$worst$. If ek is better, it replaces Gk,$worst$, resetting η to 0. Otherwise, Gk,$worst$ remains unchanged, and η is decremented by 1. This process utilizes historical information to enhance population diversity and overcome stagnation.

3.4.3 Spatial Exclusion Strategy

The goal of employing a multiple swarm strategy is to expand search areas and population variety. However, in the presence of local optima, distinct sub-swarms might easily mix, resulting in the population's premature convergence.

To overcome this issue, the Spatial Exclusion Strategy (SES) is used, which takes into account particle spatial distance and the overall evolutionary condition of the population to avoid sub-swarms from combining prematurely. Initially, sub-swarms are sent to explore various locations. With time, the geographical exclusion technique weakens, and all sub-swarms focus on local search.

The population's evolutionary progress is gauged by the count of fitness assessments. The parameter S undergoes a gradual transition from 0 to 1 as evolution unfolds. In every generation, a random value r is juxtaposed with P, prompting the application of diverse mechanisms in an adaptable manner contingent upon the outcome.

S = fes/Max Fes (34)

If the randomly generated number r is smaller than S, it indicates that the population is in the advanced stage of evolution, where local search is prioritized and the SES is not activated. Conversely, if r is larger than S, it signifies that the population is in the initial stage of evolution, prompting the activation of the SES.

The SES employs the spatial distance amongst each particle and the global best position (gbest) to determine repulsion. A larger distance results in weaker repulsion, allowing particles to explore other promising regions. Conversely, a shorter distance leads to stronger repulsion, guiding particles to focus on their respective regions.

The proposed SES adjusts spatial exclusion dynamically according to the population's evolutionary stage. This method increases population diversity during the early phases of evolution while preserving convergence precision in the later stages.

$$disti=\sqrt{\sum_{d=1}^{D}(gbest\ -x_i^d)} \quad (35)$$

$$Xid = Xid + rd *(Ud – Ld) * \exp(-disti) \quad (36)$$

In the case where a particle's position exceeds the defined boundaries, it is necessary to reinitialize the particle's position. This process is denoted by Eq. (36), where r = random number [0,1], and $disti$ signifies the spatial distance between the ith particle and gbest.

$$Xid = rand * (Ud – Ld) \quad (37)$$

3.5 Summary

Meta-heuristic optimisation algorithms are a set of strategies aimed at discovering the best solution to a problem that may involve a huge number of variables or complex constraints, allowing you to tackle scheduling difficulties with less effort. Meta-heuristic approaches are very beneficial for tackling optimisation issues with complicated constraints and objectives, where traditional techniques may be ineffective. Instead, this work proposes three new revolutionary meta-heuristics: GA, NG ABCA, Dy-PSO, and Nv-PSo. The specific operation of each algorithm is explained in the appropriate parts. The following parts of this thesis go into depth about how these algorithms were implemented for various scheduling challenges in FMS.

CHAPTER 4

FLEXIBLE JOB SHOP SCHEDULING - A MULTI-OBJECTIVE SIMULTANEOUS SCHEDULING APPROACH

4.1 Introduction

Flexible Manufacturing Systems (FMS) play a vital role in modern production environments by providing adaptability to varying product demands, reducing lead times, and optimizing resource utilization. Efficient scheduling in FMS is a significant challenge due to the presence of multiple machines, workstations, and dynamic job allocations. Traditional scheduling approaches often struggle with computational complexity and suboptimal solutions in such dynamic settings. Therefore, the need for advanced metaheuristic algorithms to optimize scheduling in FMS has become increasingly important.

In this study, a novel hybrid algorithm, NG-AABCA (Novel Genetic and Adaptive Artificial Bee Colony Algorithm), is introduced to tackle simultaneous scheduling problems in FMS. This approach integrates the strengths of the Novel Genetic Algorithm (NG) and the Adaptive Artificial Bee Colony Algorithm (AABCA) to enhance performance in solving scheduling optimization problems. By incorporating both social and cognitive learning factors, NG-AABCA strategically utilises external sources to derive superior solutions, thereby amplifying productivity and minimizing penalties. The algorithm is particularly designed to optimise makespan, total tardiness, and penalty costs in FMS, making it highly effective for complex manufacturing environments.

FMS scheduling problems require efficient resource allocation strategies to ensure smooth operations while minimising delays and costs. Metaheuristic algorithms such as Genetic Algorithms (GA) and Artificial Bee Colony (ABC) algorithms have been widely used in optimisation problems related to FMS. The GA employs evolutionary principles inspired by natural selection, allowing populations to evolve toward optimal solutions. However, GAs often suffer from premature convergence, limiting their effectiveness in reaching the global optimum. On the other hand, the ABC algorithm, inspired by the foraging behaviour of honeybees, efficiently explores the search space but may struggle with fine-tuning solutions. By combining the adaptability of NG and AABCA, the proposed NG-AABCA method leverages the advantages of both algorithms to ensure a balance between exploration and exploitation, leading to improved scheduling performance in FMS.

The efficacy of NG-AABCA is validated through comparative analysis against benchmarking problems derived from Bilge Ulsoy's 82 problems, generated from four different layouts in an FMS setting. The algorithm is tested on various job sets and layouts, demonstrating its capability to achieve significant reductions in makespan, tardiness, and

penalty costs. Experimental results indicate that NG-AABCA outperforms conventional scheduling methods, highlighting its potential as a robust optimisation approach specifically designed for FMS.

By addressing key scheduling challenges in FMS with an innovative hybrid approach, this paper adds to the field of optimisation by offering an effective and adaptive scheduling strategy. The integration of genetic and artificial bee colony principles ensures improved computational efficiency and solution quality, making NG-AABCA a promising tool for real-world FMS scheduling applications.

4.2 Flexible Job Shop Scheduling Framework

In this FMS configuration given in Figure 6, there are four CNC machines with automatic tool changers and pallet changers. This case study has 12 job sets, each processing 4 to 8 job sequences of a different order. There are four different layouts based on which, in total 82 problems are generated. There are loading and unloading stations where the unfinished jobs are assigned and finished jobs are submitted.

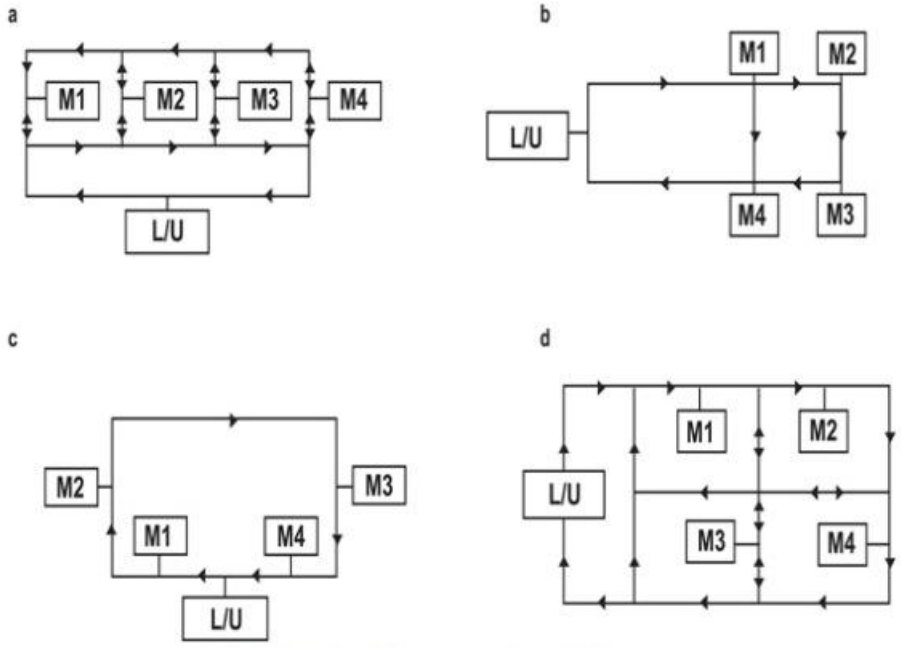


Figure 6 - Different FMS Configuration

4.2.1 Mathematical model:

The main goal is to reduce the makespan, i.e., the total duration needed to complete a set of tasks or operations.Hafidz & Murata (2010) calculated total tardiness and penalty costs associated with simultaneous scheduling, which have not met the due dates given by the customers.

It's a crucial metric in scheduling and operations management, where minimizing makespan is often the goal to ensure efficient resource utilization and timely task completion. Achieving an optimal makespan involves effective resource allocation and task sequencing to minimize idle time and meet deadlines.

The processing time of each operation ‘j’ of job ‘i’

$$Pij = Mij + Sij + Lij + ULij + Tcij, \quad (38)$$

where

Mij - Machining time

Sij-Set uptime

Lij -Loading time

ULij -Unloading time

TCij -Tool changing time

Total operation time

$$OTij = TTij + Pij, \quad (39)$$

where

TTij= travelling time

Pij= processing time.

where n = number of jobs; Ti = Tardiness: PC= Penalty Cost

With the help of equations (38) and (39), the processing time and total operation completion are calculated. The travel-to-processing time is very crucial in such a type of problem. By using the different layouts from the research, such as line, ladder, etc four layouts are generated having four machines. The travel time is given in Table 1. The loaded trip time is obtained by adding L/U time(Bilge & Ulusoy, 1995) to travel time as given in the Appendix. The major challenge is the ti/pi ratio, i.e., travel time to processing time, because it impacts how the schedules of machines and vehicles interact. Datasets are categorized into two groups based on their ti/pi ratios. Datasets with ratios greater than 0.25 are placed in the first group, while those with ratios less than 0.25 are classified into the second group.

Table 1: Travel time data of 4 different layouts in sec (Bilge & Ulusoy, 1995)

		L/U	M1	M2	M3	M4
L1	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
		L/U	M1	M2	M3	M4
L2	L/U	0	4	6	8	6
	M1	6	0	2	4	2
	M2	8	12	0	2	4
	M3	6	10	12	0	2
	M4	4	8	10	12	0
		L/U	M1	M2	M3	M4
L3	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	4	8	10	12	0
		L/U	M1	M2	M3	M4
L4	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

4.2.2 Assumptions:

The following assumptions are made.

1 Type of machines is fixed.

2 Machines no's is fixed.

3 AGV speed is 40m/min

4 The distance between the machines is known.

5 The average makespan, evenly divided throughout the jobs, is taken as the due date for the test issues.

6 Allocation of tools and operations to the machines are made.

4.3 Methodologies Proposed

The FJSSP is tried in this study by developing two unique metaheuristic techniques: NG-AABCA and unique GA. This section describes the implementation steps, as well as the encoding and decoding of NG-AABCA and Novel GA for FJSSP.

4.3.1 Implementation of NG-AABCA

The Novel Genetic and Adaptive Artificial Bee Colony (GA-ABC) Algorithm is a hybrid metaheuristic designed to enhance the efficiency and accuracy of solving complex optimization problems such as the Flexible Job Shop Scheduling Problem (FJSSP). By combining the global search capabilities of Genetic Algorithms (GA) with the local exploitation strength of an Adaptive Artificial Bee Colony (ABC) algorithm, this approach addresses the limitations of premature convergence in GA and weak exploitation in standard ABC. The genetic component ensures diversity and prevents stagnation through crossover and mutation, while the adaptive ABC dynamically adjusts its search strategy based on feedback from the environment. This synergy enables the GA-ABC algorithm to achieve a balanced exploration–exploitation trade-off, resulting in faster convergence and improved solution quality in high-dimensional and constraint-rich scheduling scenarios.

Initialise population : The implementation of the NG-AABCA (Novel Genetic and Adaptive Artificial Bee Colony Algorithm) begins with the initialization phase, where the number of food sources, limit for abandonment, and termination criteria are defined. The initial population of food sources is generated randomly within specified boundaries, and their fitness values are evaluated. In the employed bee phase, each bee selects a food source and generates a new one by making slight modifications to its current position. If the new food source has a better fitness value, it replaces the previous one; otherwise, the existing source remains, and a counter is incremented.

Following this, the onlooker bee phase evaluates the food sources and selects them based on their fitness probability. The selected food sources are further improved through adaptive modifications. If a food source is not improved for a set number of trials, it is abandoned, and a scout bee randomly generates a new food source to maintain diversity in the search space. To enhance the exploration-exploitation balance, the algorithm integrates a Novel Genetic Algorithm (NG) component, which injects high-quality solutions into the population and prevents stagnation at local optima. Crossover and mutation operators are applied to improve solution diversity.

The dynamic learning factors incorporated in the algorithm help balance global exploration and local exploitation by adjusting their influence based on the number of iterations. This ensures that the algorithm initially focuses on exploring a wide range of solutions before converging toward optimal results. The algorithm continues executing these phases iteratively until the termination criteria are met, ultimately providing an optimized solution that minimizes makespan, tardiness, and penalty costs in scheduling problems. The integration of NG with AABCA enhances adaptability, convergence speed, and solution quality, making it an efficient approach for solving complex scheduling challenges in

flexible manufacturing systems.

4.3.2 Solution Representation

In the proposed GA-ABC algorithm, each chromosome encodes job permutations focusing on operation sequences and corresponding machine assignments. The representation ensures adherence to job precedence and machine constraints by mapping operations (e.g., O21, O11) to specific machines (e.g., M1, M5) through indexed values. Crossover and mutation operators are employed to explore the solution space: single-point crossover generates valid offspring by combining parent sequences, while swap mutation introduces diversity by interchanging genes, preventing local optima entrapment. To maintain high-quality solutions across generations, elitism is applied—preserving the best chromosomes based on fitness (e.g., total operation time or mean tardiness) to accelerate convergence without losing optimal candidates. This integrated representation supports robust scheduling in FJSSP environments.

4.4 Input Test Instances

The main goal is to reduce the makespan, i.e., the total duration needed to complete a set of tasks or operations. 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. *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.
2. *Bilge Ulsoy data set :* This data set consists of **0 job sets** and **4 distinct manufacturing layouts**, resulting in a total of **40 unique instances.**

4.5 Proposed Algorithms Validation and Discussions

4.5.1 Brandimarte's Data set [BR Data Set]

The **Brandimarte (BR) dataset**, introduced in 1993, is a widely used benchmark for Flexible Job Shop Scheduling Problems (FJSSP). It includes **10 problem instances** with varying numbers of jobs and machines, incorporating machine flexibility and operation sequencing. The dataset is commonly used to evaluate metaheuristic algorithms in complex scheduling environments.The research involved the development and evaluation of a new algorithm called NG-AABCA (Novel Genetic- Adaptive Artificial Bee Colony Algorithm). The purpose of algorithm was to solve a particular problem related to job scheduling and optimization, where the objective was to minimize the makespan of a given set of jobs. In this research, the performance of NG-AABCA with seven other algorithms that were previously proposed in the literature are compared. The evaluation was conducted using MATLAB R2019b, a widely used programming environment for numerical computations and algorithm development. The algorithms were tested on various job sets, each presenting a different level of difficulty or complexity, as stated in Table 2. The primary metric used to assess the algorithms' performance was the makespan

value, using equations (38) and (39) to achieve the minimum possible makespan and tardiness for each job set. The results of the evaluation demonstrated that NG-AABCA consistently outperformed the other seven algorithms across different job sets of Layout 1. In the context of the optimization process, several key procedural parameters are defined. These parameters encompass a maximum population size of 500, 30 specified iterations, a crossover probability (pc) set at 0.8, a mutation probability (pm) of 0.2, and two epsilon values (ε1 and ε2) established at 0.8 and 0.5, respectively. These parameters play a pivotal role in guiding the genetic algorithm's search for the optimal solution, striking a steadiness between exploration and exploitation while ensuring convergence in addition to adaptability to the problem space.

Table 2. Calculation of makespan using NG-AABCA

CALCULATION OF MAKESPAN								
Sr.no	J	M/C	AGV NO	TRAVEL TIME	JOB REACH	JOB READY	PROCESS TIME	JOB COMPLETION
1	1	1	1	0	6	6	15	21
2	3	4	2	0	6	26	6	32
3	11	2	1	18	30	30	12	42
4	4	1	2	18	28	28	10	38
5	5	3	1	36	46	46	8	54
6	8	4	2	36	42	42	12	54
7	10	4	1	46	52	52	7	59
8	7	3	2	48	56	56	8	64
9	2	2	2	56	64	64	12	76
10	12	3	1	52	60	60	15	75
11	13	1	1	60	68	68	10	78
12	6	2	1	68	78	79	12	91
13	9	1	2	72	78	78	16	94

The program was executed using Novel GA and Adaptive ABC algorithms, with efforts made to optimize their use. After 10 repetitions, the best results were obtained. The NG-AABCA algorithm generated the best sequence, which is. resulting in a total makespan of 94.

4.5.2 Bilge Ulusoy data set:

Bilge and Ulusoy (1995) developed a benchmark dataset comprising **10 job sets** and **4 distinct manufacturing layouts**, resulting in a total of **40 unique instances** for the Flexible Job Shop Scheduling Problem with Transportation (FJSPT). These instances have been widely adopted in the research community to evaluate and compare various scheduling algorithms. Each instance incorporates considerations for machine assignments, operation sequences, and transportation constraints, making them particularly suitable for studying complex scheduling scenarios in flexible manufacturing systems.

The job sets were categorized according to the t/p ratio, where t represents tardiness and p represents processing time. Tables (3) and (4) given below depict the findings of this investigation, which indicate that NG-AABCA produced a better optimization solution. This algorithm is particularly useful for the

simultaneous scheduling of machines and AGVs.

In Table 3 t/p ratio is greater than 0.25, which indicates that the tardiness (t) of the job is relatively high compared to its processing time (p). So, it suggests that the job is taking significantly longer to complete than originally scheduled or desired, resulting in a delay or tardiness that exceeds 25% of its processing time.

The number of jobs that have a greater t/p ratio is less in the case of NG-AABCA, which proves the efficacy of the algorithm. NG-AABCA continued to excel and surpassed algorithms such as Sliding Time Window (STW), Ulusoy Genetic Algorithm (UGA), Abdelmaguid Genetic Algorithm (AGA), and In scenarios with t/p greater than 0.25, NG-AABCA's performance was on par with, Rao And Reddy Genetic Algorithm (PGA), Deroussi Hybrid Algorithm (DHA), Chowdary Genetic Algorithm (CGA), Hybrid Genetic Vehicle Heuristic Algorithm (HGVHA).

Table 3. Performance Evaluation of Job Tardiness (t/p>0.25)

P.No	t/p	STW	UGA	AGA	PGA	DHA	CGA	HGVHA	NG-AABCA
1.1	0.59	0	0	0	0	0	0	0	0
2.1	0.61	5	4	2	0	0	0	0	0
3.1	0.59	6	6	0	0	0	0	1	1
4.1	0.91	6	4	0	0	0	0	2	1
5.1	0.85	2	0	0	0	0	0	0	0
6.1	0.78	2	3	0	0	0	0	0	0
7.1	0.78	8	7	4	0	0	4	0	1
8.1	0.58	9	0	9	9	9	9	9	8
9.1	0.61	4	1	2	0	0	0	0	0
10.1	0.55	6	3	0	0	0	3	1	0
1.2	0.47	0	0	0	0	0	0	0	0
2.2	0.49	4	0	0	0	0	0	0	0
3.2	0.47	3	0	0	0	0	0	0	0
4.2	0.73	7	2	2	1	1	2	0	1
5.2	0.68	0	0	0	0	0	0	0	0
6.2	0.54	2	0	0	0	0	0	0	0
7.2	0.62	11	6	0	0	0	2	0	0
8.2	0.46	9	0	9	9	9	9	9	9
9.2	0.49	2	0	2	0	0	0	0	0
10.2	0.44	4	2	1	0	0	6	0	0
1.3	0.52	0	0	0	0	0	0	0	0
2.3	0.54	0	0	0	0	0	0	0	0
3.3	0.51	0	0	0	0	0	0	0	0
4.3	0.8	6	2	0	0	0	0	0	0
5.3	0.74	2	1	0	0	0	0	0	0
6.3	0.54	1	1	1	0	0	1	0	0
7.3	0.68	9	6	4	1	4	8	0	0
8.3	0.5	10	0	10	10	10	10	10	8

9.3	0.53	5	0	1	0	0	0	0	0
10.3	0.49	5	5	3	1	0	2	1	1
1.4	0.74	5	0	0	0	0	0	0	0
2.4	0.77	8	5	0	0	0	0	0	0
3.4	0.74	6	3	1	1	1	1	0	0
4.4	1.14	5	5	5	5	0	5	5	5
5.4	1.06	3	1	0	0	0	0	0	0
6.4	0.78	0	3	0	0	0	2	0	0
7.4	0.97	10	2	1	0	0	4	0	0
8.4	0.72	0	0	0	0	0	0	0	0
9.4	0.76	5	3	2	2	0	0	2	2
10.4	0.69	13	6	1	0	1	1	1	0

Table 4. Performance Evaluation of Job Tardiness (t/p<0.25)

P.No	t/p	STW	UGA	AGA	PGA	DHA	CGA	HGVHA	NG-AABCA
1.1	0.15	0	0	0	0	0	0	0	0
2.1	0.15	0	0	0	0	0	0	0	0
3.1	0.15	2	0	2	2	2	2	2	2
4.1	0.15	2	0	0	0	0	0	0	0
5.1	0.21	0	0	0	0	0	0	0	0
6.1	0.16	0	0	0	0	0	0	0	0
7.1	0.19	0	0	0	0	0	0	0	0
8.1	0.14	21	0	21	21	21	21	21	20
9.1	0.15	0	0	0	0	0	0	0	0
10.1	0.14	2	0	2	2	2	2	2	2
1.2	0.12	0	0	0	0	0	0	0	0
2.2	0.12	0	0	0	0	0	0	0	0
3.2	0.12	3	0	0	0	0	0	0	0
4.2	0.12	2	0	0	0	0	0	0	0
5.2	0.17	0	0	0	0	0	0	0	0
6.2	0.12	2	0	0	0	0	0	5	0
7.2	0.15	0	0	0	0	0	0	0	0
8.2	0.11	19	0	19	19	19	19	19	18
9.2	0.12	1	0	0	0	0	0	0	0
10.2	0.11	0	2	0	0	0	0	0	0
1.3	0.13	0	0	0	0	0	0	0	0
2.3	0.13	0	0	0	0	0	0	0	0
3.3	0.13	3	0	0	0	0	0	0	0
4.3	0.13	2	0	0	0	0	0	0	0
5.3	0.18	0	0	0	0	0	0	0	0
6.3	0.24	2	0	0	0	0	0	0	0
7.3	0.17	0	0	0	0	0	0	0	0
8.3	0.13	18	0	18	18	18	18	18	17

9.3	0.13	2	0	0	0	0	0	0	0
10.3	0.12	0	4	0	0	4	0	0	0
1.4	0.18	0	0	0	0	0	0	0	0
2.4	0.13	0	0	0	0	0	0	0	0
3.4	0.18	0	0	0	0	0	0	0	0
3.4	0.12	1	0	0	0	0	0	0	0
4.4	0.19	8	1	1	1	8	1	0	0
5.4	0.18	6	0	0	0	0	0	0	0
6.4	0.19	1	0	0	0	0	0	0	0
7.4	0.24	1	0	0	0	0	0	0	0
7.4	0.16	0	0	0	0	0	0	0	0
8.4	0.18	20	0	20	20	20	20	20	17
9.4	0.19	2	0	0	0	0	0	0	0
10.4	0.17	0	4	0	0	0	0	0	0

In the above table, the t/p ratio is less than 0.25, which indicates that the tardiness (t) of the job is relatively low compared to its processing time (p). In other words, it suggests that the job is being completed well within its scheduled or desired time, and there is minimal delay or tardiness. Specifically, the tardiness is less than 25% of the processing time, which implies that the job is typically finished well before its expected completion time. In more than 80% of the tested scenarios, NG-AABCA showed strong performance, achieving the minimum makespan values. Specifically, when compared to algorithms like STW, UGA, IACGA, HGVHA, DGA, and STW, NG-AABCA achieved the shortest relative makespan for certain conditions, particularly when t/p (Tardiness of job/ Processing time of job) was 0.25.

Table 5. Performance Comparison of Job Tardiness

P.No	UGA			GAA			SALS			PGA			Proposed GA				NG-AABCA		
		A	B		A	B		A	B		A	B		A	B			B	
	Result	%	%	Result	%	%	Result	%	%	Result	%	%	Result	%	%	Time	Best	%	Time
1.1	96	0	0	96	0	0	96	0	0	96	0	0	96	0	0	7	95	0	5
2.1	82	0	0	82	0	0	82	0	0	82	0	0	82	0	0	6	80	0	4
3.1	84	0	0	84	0	0	84	0	0	84	0	0	84	0	0	18	82	0	15
4.1	103	0	0	103	0	0	103	0	0	103	0	0	103	0	0	23	103	0	20
5.1	104	4	4	102	2	2	100	0	0	100	0	0	100	0	0	113	98	0	100
6.1	76	0	0	76	0	0	76	0	0	76	0	0	76	0	0	18	75	0	17
7.1	86	0	0	86	0	0	86	0	0	86	0	0	86	0	0	7	85	0	6
8.1	113	4.63	4.63	108	0	0	108	0	0	108	0	0	108	0	0	11	105	0	10
9.1	105	6.061	6.06	99	0	0	99	0	0	99	0	0	99	0	0	34	95	0	25
10.1	85	0	0	85	0	0	85	0	0	85	0	0	85	0	0	15	85	0	14
1.2	86	0	0	86	0	0	86	0	0	86	0	0	86	0	0	54	83	0	47

2.2	113	1.802	1.8	111	0	0	111	0	0	111	0	0	111	0	0	18	111	0	17
3.2	116	3.571	3.57	112	0	0	112	0	0	112	0	0	112	0	0	193	107	0	185
4.2	88	1.149	1.15	88	0	1.15	87	−1.136	0	87	−1.136	0	88	1.149	1.15	54	85	1	53
5.2	91	2.247	2.25	89	0	0 0	89	0	0	89	0	0	89	0	0	25	87	0	24
6.2	126	4.132	4.13	126	0	4.13	121	−3.968	0	126	0	4.1	126	4.132	4.13	68	125	3.1	60
7.2	87	0	0	87	0	0	87	0	0	87	0	0	87	0	0	68	85	0	67
8.2	69	0	0	69	0	0	69	0	0	69	0	0	69	0	0	10	67	0	9
9.2	75	1.351	1.35	74	0	0	74	0	0	74	0	0	74	0	0	45	74	0	40
10.2	97	1.042	1.04	96	0	0	96	0	0	96	0	0	96	0	0	29	94	0	28
1.3	121	2.542	2.54	118	0	0	118	0	0	118	0	0	118	0	0	126	115	0	112
2.3	98	0	0	98	0	0	98	0	0	98	0	0	98	0	0	12	95	0	11
3.3	104	0	0.97	104	0	0.97	103	−0.962	0	103	−0.962	0	104	0.971	0.97	120	102	0.7	115
4.3	123	0.82	2.5	120	−1.639	0	120	−1.639	0	120	−1.639	0	122	1.667	1.67	12	122	1	11
5.3	118	2.609	6.31	115	0	3.6	111	−3.478	0	111	−3.478	0	115	3.604	3.6	104	115	2.604	85
6.3	85	4.938	7.6	79	−2.469	0	79	−2.469	0	79	−2.469	0	81	2.532	2.53	24	80	0	23
7.3	88	-2.22	6.02	86	−4.444	3.61	83	−7.778	0	83	−7.778	0	90	8.434	8.43	30	88	1	23
8.3	128	-1.54	1.59	127	−2.308	0.79	126	−3.077	0	126	−3.077	0	130	3.175	3.18	32	125	0	25
9.3	161	0	0	161	0	0	161	0	0	161	0	0	161	0	0	13	158	0	12
10.3	151	0	0	151	0	0	151	0	0	151	0	0	151	0	0	4	150	0	3
1.4	153	0	0	153	0	0	153	0	0	153	0	0	153	0	0	5	148	0	4
2.4	163	0	0	163	0	0	163	0	0	163	0	0	163	0	0	38	160	0	31
3.4	117	0.862	0.86	118	1.724	1.72	116	0	0	116	0	0	116	0	0	41	113	0	40
4.4	102	0	0	104	1.961	1.96	102	0	0	102	0	0	102	0	0	20	97	0	19
5.4	105	0	0	106	0.952	0.95	105	0	0	105	0	0	105	0	0	21	103	0	20
6.4	123	2.5	2.5	122	1.667	1.67	120	0	0	122	1.667	1.7	120	0	0	42	116	0	41
7.4	150	0	2.04	147	−2.00	0	147	−2	0	147	−2.000	0	150	2.041	2.04	75	143	1	72
8.4	137	-2.84	1.48	136	−3.546	0.74	135	−4.255	0	135	−4.255	0	141	4.444	4.44	30	138	2	29
9.4	143	2.143	3.62	141	0.714	2.17	138	−1.429	0	139	−0.714	0.7	140	1.449	1.45	83	139	0	77
10.	164	3.14	3.8	159	0	0.6	159	0	0.	158	−0.6	0	159	0.63	0.6	161	15	0	142

4	5	3	6	29	3	3	7

Table 5 presents an analysis of the performance evaluation between the proposed algorithm and various other algorithms, as well as a performance comparison between the best algorithm and those same alternative algorithms. Notably, the results indicate that when all these algorithms are compared with NG-AABC, they consistently exhibit a tardiness rate that is approximately 6% to 10% higher than that observed in UGA, DGA, PGA, and the novel GA. The other notable advantage of NG-AABCA was its significantly reduced computational time compared to the Proposed GA under evaluation. The computational efficiency of an algorithm is influenced by factors like the algorithm's design, the specific implementation, and the hardware it's run on. This research proves that the proposed techniques consistently demonstrated greater performance in terms of computational efficiency. Despite the influence of the operating system, their techniques managed to complete tasks within seconds. The experiments and tests were carried out using MATLAB versions from 2021 onwards, and the testing environment consisted of computers equipped with Intel Core i5 2.3-GHz CPUs.

In summary, the study introduced a novel algorithm, NG-AABCA, for solving a specific problem. Through comprehensive evaluations, the algorithm showcased its superiority over existing algorithms in terms of both make span optimization and computational efficiency.

Table 6. Comparison of outcomes in relation to NG-AABC

Method	Same	Better	Worse
Comparison with UGA	5	35	0
Comparison with GAA	8	32	0
Comparison with DGA	13	27	1
Comparison with PGA	13	26	1

Table 7. Comparison of outcomes with the best solution for all four algorithms

UGA		GAA		DGA		PGA		NG-AABCA	
Same	Worse	Same	Worse	Same	Worse	Same	Worse	Same	Worse
15	25	18	22	30	10	34	6	36	4

Table 6 presents the outcomes for the four aforementioned methods: UGA, GAA, DGA, and PGA. PGA, alongside NG-AABCA, applied to problems with t_i/p_i ratios exceeding 0.25, totaling 40 problems. This table illustrates how the NG-AABCA performs in relation to the four previous approaches (UGA, GAA, DGA, and PGA), indicating the percentage deviation for each. Additionally, it provides the percentage deviation compared to the best-known solution obtained by each of the five approaches for these 40 problems. Table 6 reveals that when compared to UGA, NG-AABCA achieved identical solutions for 5 problems, outperformed it in 35 problems, and fell short in producing better results for no problems. In contrast, when compared to GAA, the NG-AABCA achieved the same solution in 8 cases

and improved upon it in 32 cases. Among the four methods presented, DGA and PGA demonstrated the highest performance. When comparing the NG-AABCA to DGA, it matched the solution for 13 problems but produced inferior results for 1 problem. In comparison to PGA, the NG-AABCA achieved identical solutions in 13 cases, improved upon it in 26 cases, and performed worse in 1 problem.

Table 7 summarizes the results in comparison to the best-known solution for all four approaches. It also provides a comprehensive visual representation of the percentage deviation, showing how the NG-AABCA compares to other approaches.

4.5.3 Experiential investigation of Impacts of parameter settings on NG-AABCA performance.

For each combination of parameters, NG-AABCA is executed with 500 distinct initial populations, ensuring a total of 30 runs. The selected values for population size encompass a typical range found in the literature, with the addition of a value of 0.2 to emphasize the impact of a relatively high mutation rate. Likewise, the chosen crossover rates span the entire spectrum. It's worth noting that the known optimal (target) solution for this schedule is a make span of 94 units of time.

Table 8. Comparative analysis of different parameters

Pop size	Mutation rate			
	0.003	0.006	0.009	0.018
20	82	77.38	77.52	81.14
50	78.9	77.14	77.38	83.36
80	77.96	78.72	78.4	83.66
Pop size	Crossover rate			
	0.2	0.35	0.65	0.8
20	80.05	79.08	79.8	79.98
50	78.75	79.08	79.1	79.5
80	79.43	79.38	79.9	80.13
Crossover rate	Mutation rate			
	0.003	0.006	0.009	0.018
0.2	79.77	77.7	77.6	82.57
0.35	78.67	77.53	77.9	82.6
0.65	80.13	77.73	77.73	82.83
0.8	80.73	77.83	77.87	83.03
	ε_2			
ε_1	0.5	0.6	0.7	0.8
0.5	98	96.3	94	92.8
0.6	96.4	95.8	94.1	93.5
0.7	97.3	96.4	95.7	95
0.8	99.8	98	97.5	97

This study systematically documents the mean makespan outcomes obtained from 500 iterations across different parameter combinations, as presented in Table 8. These averages are derived from diverse initial populations and varying levels of the third parameter. To simplify, each entry in Table 8 represents the averaged result of 500 individual runs, which stem from the combination of five distinct crossover rate levels and various early populations. In terms of the mean makespan values obtained, Table 8 provides valuable insights into performance trends. Significantly, the study observes a performance decrease when both the population size and mutation rate are set to minimum values. This indicates that when the population size is exceedingly small, a substantial increase in the mutation rate is necessary to pay off for the limited occurrence of mutations. It's important to note that the mutation rate's frequency is influenced by both the population size and the mutation rate itself. Furthermore, a smaller population inherently lacks diversity and is more prone to being dominated by local optimal

solutions. Reducing the mutation rate exacerbates this issue, causing the genetic algorithm (GA) to allocate more time to exploring localized optimal regions.

On the contrary, when the population size is substantial, a lower mutation rate proves to be more efficient. This can be attributed to the proportional principle, which suggests that a larger population offers greater space for diversity. Nevertheless, it's essential to acknowledge that performance across the range of population sizes deteriorates when the mutation rate is excessively high. It is because an overly elevated mutation rate causes the population to navigate the search space erratically, diverting attention from a thorough exploration of optimal regions. Essentially, in such scenarios, the mutation operator takes precedence over the crossover operator, resulting in suboptimal performance. The experiment involves ten distinct initial populations, resulting in a total of 500 runs. The selected population size values cover a typical range commonly encountered in existing literature.

Additionally, a deliberate inclusion of a mutation rate value of 0.018 underscores the impact of a high mutation rate. The chosen crossover rates span the entire spectrum. It's important to emphasize that the known optimal (target) solution for this scheduling problem is a makespan of 94 units of time.

4.6 Summary

This study extends its analysis to record the mean makespan achieved after 500 iterations for each parameter combination outlined in Table 8. These mean values are computed across different levels of the corresponding third parameter and varied initial populations. To clarify, each entry in Table 8 represents the average result of 50 individual runs. Once again, in terms of the mean makespan attained, Table 6 provides insights into performance trends, echoing the observations made in Table 8 concerning the impact of population size and mutation rate on the algorithm's effectiveness. In summary, the study diligently explores the consequences of altering population size and mutation rates on the performance of a GA, shedding light on the delicate equilibrium required to attain optimal results in solving the scheduling problem at hand.

The cognitive learning factor (ε_1) represents the extent to which an individual bee relies on its own knowledge or memory of previously visited solutions when making a decision. In the context of the algorithm, ε_1 is used to adjust the candidate solutions generated by an individual bee based on its own experience. It helps the bee decide how much it should explore around its previously visited solutions. A higher value of ε_1 would make the bee rely more on its own experience and focus on exploring nearby solutions, potentially exploiting known good areas of the solution space.

Conversely, a lower value of ε_1 would make the bee more exploratory and less reliant on its own memory, encouraging it to explore a wider range of solutions. The social learning factor (ε_2) represents the extent to which an individual bee communicates and learns from the experiences of other bees in the colony. In the ABC algorithm, ε_2 is used to adjust the candidate solutions generated by an individual bee based on information gathered from other bees in the colony. It helps the bee decide how much it should explore solutions that are discovered by its peers. A higher value of ε_2 would make the bee rely more on information from other bees, potentially leading to a more collective or cooperative search behavior

where the bee explores solutions discovered by its peers. A lower value of ε_2 would make the bee less influenced by the experiences of other bees, promoting a more individualistic search strategy. The balance between ε_1 and ε_2 is crucial in the ABC algorithm. A higher ε_1 value may encourage the exploitation of known good solutions, while a higher ε_2 value may encourage exploration and the sharing of information within the bee colony. The appropriate values for ε_1 and ε_2 depend on the specific optimization problem being solved and may require tuning through experimentation to achieve the best results. So, overall, ε_1 and ε_2 in the ABC algorithm represent the trade-off between individual learning (exploitation) and collective learning (exploration) in the context of a bee colony searching for optimal solutions. The optimal values of ε_1 and ε_2, which yield the minimum makespan values, are 0.8 & 0.5. These factors are essential for achieving a balance between exploration and exploitation during the search procedure.

Chapter 5

Simultaneous Scheduling–Layout Optimization for Minimizing Total Energy Consumption in Flexible Manufacturing Systems

5.1 Introduction:

In modern manufacturing environments, improving energy efficiency requires a holistic consideration of both production scheduling and facility layout design. This chapter addresses the problem by extending the classical Bilge and Ulusoy (1995) simultaneous scheduling and layout model to incorporate machining tool data and machine energy consumption states. The study evaluates 82 problem instances derived from 10 job sets and four travel-time matrices, integrating transportation energy consumption with machine operating modes—processing, brief dormant, and extended dormant. Material flow efficiency, measured through Total Energy Consumption (TEC), is influenced not only by the allocation and sequencing of operations across multiple machines but also by the spatial arrangement of these machines. By capturing the interaction between scheduling decisions and physical layout, this work provides a more realistic representation of manufacturing systems, where job transportation times, machine idle thresholds, and energy use are interdependent factors shaping overall system performance. Simultaneous scheduling focuses on jobs, m/c's, tools, and Automated Guided Vehicles. Each job's operation has a set processing time, and each operation can be completed on at least one machine. To facilitate processing, an AGV is necessary to carry each work to the designated machine, with transportation time varying based on the route and direction. Initially, all jobs and AGVs start at the warehouse. Within the scope of the Multi-Objective Green Scheduling Problem, four interconnected sub-problems must be optimised simultaneously.

Machine Assignment (MA): Designating a processing machine for each operation.

Operation Sequence (OS): Establishing the order of operations for each machine.

AGV Selection: Choosing an AGV for each operation.

Task Sequence (TS): Planning the order of transportation tasks for each AGV.

The idea is to reduce both the makespan and overall energy consumption concurrently. The problem is governed by various assumptions. All jobs and machinery are available from the beginning. Each machine can only perform one action at a time. Operations cannot be preempted. Precedence relationships only apply within the same employment. The transit time includes both loading and unloading periods. Each AGV only carries one operator at a time.

Table 9: Transportation time

		L/U	M/C1	M/C2	M/C3	M/C4
L1	L/U	0	6	8	10	12
	M/C1	12	0	6	8	10
	M/C2	10	6	0	6	8
	M/C3	8	8	6	0	6
	M/C4	6	10	8	6	0
		L/U	M/C1	M/C2	M/C3	M/C4
L2	L/U	0	4	6	8	6
	M/C1	6	0	2	4	2
	M/C2	8	12	0	2	4
	M/C3	6	10	12	0	2
	M/C4	4	8	10	12	0
		L/U	M/C1	M/C2	M/C3	M/C4
L3	L/U	0	2	4	10	12
	M/C1	12	0	2	8	10
	M/C2	10	12	0	6	8
	M/C3	4	6	8	0	2
	M/C4	4	8	10	12	0
		L/U	M/C1	M/C2	M/C3	M/C4
L4	L/U	0	4	8	10	14
	M/C1	18	0	4	6	10
	M/C2	20	14	0	8	6
	M/C3	12	8	6	0	6
	M/C4	14	14	12	6	0

Table 10: Processing Time

Jobs	Operation I	Operation II	Operation III
Job 1 (J_1)	M1-T3(8)	M2-T4(16)	M4-T1(12)
Job 2 (J_2)	M1-T2(20)	M3-T3(10)	M2-T1(18)
Job 3 (J_3)	M3-T1(12)	M4-T4(8)	M1-T2(15)
Job 4 (J_4)	M4-T3(14);	M2-T4(18)	---
Job 5 (J_5)	M3-T2(10);	M1-T1(15)	---

The test scenarios are created based on 10 sets of jobs outlined in the Appendix and involve four matrices representing travel times. As a result, a total of 82 problems have been generated following the specifications outlined by Bilge and Ulusoy (1995). It's important to note that, unlike the original problem by Bilge and Ulusoy, the present study incorporates data related to machining tools.

In manufacturing systems, the efficiency of material transportation is evaluated through Total Energy Consumption (TEC), calculated from the total transportation time and the average power required by material handling equipment. This efficiency is heavily influenced by the material flow design, which dictates the progression of jobs across the shop floor in line with scheduling decisions. In simultaneous scheduling, each job set comprises multiple operations, with each operation assigned to a specific machine chosen from several alternatives. The scheduling process therefore entails both the allocation of operations to machines (operation assignment) and the determination of their execution order on each

machine (operation sequencing). These decisions have a direct impact on material flow. Additionally, the spatial arrangement and positioning of machines play a crucial role in ensuring efficient material movement, aiming to reduce the distance and time needed for jobs to travel between machines. As a result, scheduling and layout interact closely, jointly shaping material flow and the associated transportation TEC.

This study introduces three machine modes to model machine Total Energy Consumption (TEC): processing mode, brief dormant mode, and extended dormant mode. The processing mode occurs when a machine is actively engaged in a job, with specific processing times and power determined by the scheduling. The layout, which is closely linked to scheduling through transportation time, also influences the machine's processing mode. Machines switch to extended dormant mode if the idle time between consecutive job operations exceeds a certain threshold; otherwise, they remain in brief dormant mode. Dormant time is the period during which a machine remains idle, awaiting the next job. The start and finish times of jobs on a machine are determined by the assignment and sequencing of operations established during scheduling. Moreover, transportation time—determined jointly by scheduling and layout—also influences when a process begins. In essence, the combination of scheduling and layout decisions governs the various dormant states a machine may enter, each consuming different levels of power.

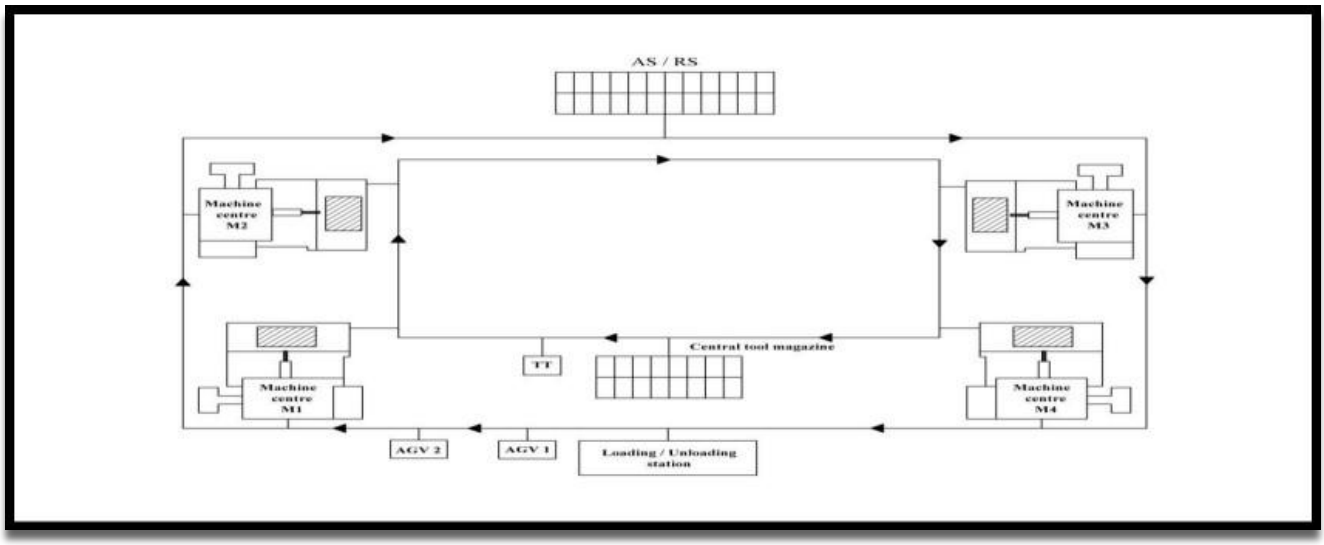


Figure 7: Layout of Machine, AGV and AS/RS

The interplay between transportation and processing tasks—especially the sequencing of transportation jobs on AGVs and tools—adds to the complexity of the problem. This complexity increases further when multiple AGVs are involved, as it requires the distribution of transportation tasks among them.

5.2 Mathematical Formulation

Assumptions

- Each machine can only process one operation at a time and cannot be interrupted once started.
- A machine starts working when a new job is ready for processing.
- The machine consumes less power during extended dormancy compared to brief dormancy.
- Machines are assigned to specific positions with predetermined spacing between them.

a∈ {1, 2, .., A} Index of jobs

b ∈ {0, 1, 2, … , B} Index of operations for job a

c ∈ {1, 2, .., C} Index of machines

d∈ {1, 2, .., D} Index of processes on machine

e∈ {1, 2, .., E} Index of location

m ∈ {1, 2, M} Index of modes of AGV i.e 1: processing mode, 2: brief dormant mode, 3: extended dormant mode

Cj Tardiness penalty of job j per unit time (cents/min)

Ce Energy cost per unit time (cents/(W-min))

W is a symbol indicating Watts, which is the unit of power

DT.: Distance between locations I and I (meter)

Dj; Due date of job j (min)

ti,j.m; Processing time of operation i of job j on machine m (min)

Time threshold of machine m (min)

Transportation time of job j per unit distance (min/meter)

aij.m 1 if operation i of job j can be processed on machine m; 0 otherwise (ajj.m is 1 if tij.m > 0; 0 otherwise)

PDPLj. Power of processing operation i of job j on machine m in states = 1 (W)

PDIm,s Power of machine m in idle states = 2 or s = 3 (W)

PDT Average power of transportation equipment (W)

H A big positive number

y 1 if machine e is assigned to location l; 0 otherwise

$\dot{x}_{dc}^{ab}$ 1 if it operation of job a is processed at d' process of machine c

Decision variables

$$C_e \sum_{d=1}^{D} \sum_{c=1}^{3} E\, CM_{d,c} + ECT \quad (40)$$

$$\sum_{e=1}^{E} y_{c,e} = 1 \quad (41)$$

$$\sum_{d=1}^{D} \quad \sum_{c=1}^{C} \quad \dot{x}_{dc}^{ab} \quad = 1 \quad (42)$$

$$\sum_{d=1}^{D} x_{dc}^{ab} \leq a_{a,b,c} \quad (43)$$

STMd,c ≥ CTMd-1,c

STMd,c ≥ $\sum_{e=1}^{E} \sum_{a=1}^{A} \sum_{\tilde{d}=1}^{D} \sum_{\tilde{c}=1}^{C} x_{dc}^{ab} \cdot X_{\tilde{d},\tilde{c}}^{b-1,a} \cdot (CTM_{\tilde{d},\tilde{c}}) + \sum_{e}^{E} \sum_{\tilde{e}\neq e} Y_{c,e} \cdot Y_{\tilde{c}\tilde{e}} \cdot D\tilde{e}\tilde{e} \cdot tr_{\dot{a}}$

(5

$$CTMd,c = STMd,c + \sum_{b=J}^{B} \sum_{a=1}^{A} x_{d,c}^{a,b} \cdot t_{a,b,c} \quad (44)$$

$$CTa \quad = \sum_{d=j}^{D} \sum_{c=1}^{C} x_{d,c}^{a,b} \cdot CTM_{b,c} \quad (45)$$

$$ECMc,m = \sum_{b=1}^{B} \sum_{a=1}^{A} \sum_{d=1}^{D} PDP_{a,b,c} \cdot \; x_{d,c}^{a,b} \left(CTM_{d,c} - STM_{d,c}\right) \quad (46)$$

$$\sum_{m=2}^{3} z_{d,c,m} = \sum_{b=1}^{B} \sum_{a=1}^{A} x_{d,c}^{a,b} \quad (47)$$

$$(STMd,c- CTMd-1,c) \leq \tau\ c + H(1+ Zd,c,3\ - \sum_{b=J}^{B} \sum_{a=1}^{A} x_{d,c}^{a,b}\ ) \quad (48)$$

$$Tc\ \leq (STMd,c- CTMd-1,c) + H(1+ Zd,c,2\ - \sum_{b=J}^{B} \sum_{a=1}^{A} x_{d,c}^{a,b}\ ) \quad (49)$$

$$ECMc,m = PDI\ c,m\ .\ \sum_{d=2}^{D} Z_{d,c,m}\ .\ (STMd,c- CTMd-1,c) \quad (50)$$

ECT= PDT X

$$\sum_{b=2}^{B} \sum_{a=1}^{A} \sum_{c=1}^{C} \sum_{\tilde{c}\neq c}^{0} \sum_{e=1}^{E} \sum_{\tilde{e}\neq 1} (\sum_{d=1}^{D} x_{d,c}^{a,b}) . \sum_{d=1}^{D} X_{\tilde{d}\tilde{c}}^{b\cdot 1,a}\ )$$

.Yc,b. $Y\tilde{d}\tilde{c}\ .D_{\tilde{e},e}\ .tra$ (51)

5.3 Parameter Values for Case Studies

The parameters for the model are established as follows.

:

1. Index of modes of AGV i.e
   - processing mode,
   - brief dormant mode,
   - extended dormant mode
2. The average power for AGV is supposed to be 700W.
3. The time threshold for all machines (m) is assumed to be 0.2 min.
4. The energy unit cost Ce is assumed to be 20 cents per kilowatt-hour.
5. The brief dormant mode (s=2), the machine power PDIm,2 is randomly generated using the uniform distribution U[300, 600] in Watts (W).
6. The extended dormant mode (s=3), the power for all machines PDIm,3 is assumed to be zero W.
7. The job tardiness penalties Cj are also randomly generated, with U[20, 30] in cents.
8. Machining of same material is considered for both algorithms to reduce the complexity.

`

5.4 Results and Discussions

To assess the efficiency of proposed PSO models for scheduling and energy minimization, we utilize a set of 13 diverse Jobsets. These datasets are designed to present a variety of challenges due to their differing dimensionalities and complexities, making them suitable for testing the robustness of our approach in simultaneous scheduling tasks. The studies were carried out using a machine powered by an Intel i7-12700K processor clocked at 3.6 GHz, 16 GB DDR4 RAM, and an NVIDIA RTX 3090 GPU. The implementation was carried out in MATLAB R2022b, with libraries like NumPy, SciPy, and TensorFlow (where needed). The operating system utilised was 04. This computational architecture provides consistency and reproducibility of the proposed NvPSO algorithm. We evaluated the performance of these PSO variations based on three main metrics: Makespan, energy usage, and detailed statistical analysis.To compare our PSO variations, we used two classic search algorithms: Artificial Immune System (AIS) and Modified Genetic Tabu Algorithm (MGTA). To ensure a fair comparison, all

approaches are evaluated using the same stopping criterion, as decided by the Makespan evaluations. In our studies, we adjusted both the population size and the maximum number of iterations to accurately match the difficulty of the task and the capabilities of the comparison algorithms.

5.4.1 Parameter Settings:

The WaOA was carried out in twenty independent runs, each consisting of a thousand iterations (T=1000). In this study, the parameter N for the WaOA is set to 20. Taking into account the computational complexity, the total number of function evaluations for each metaheuristic algorithm comes to 60,000. To validate we solved a small problem using the MATLAB solver. This case study involved four machines and 13 jobsets each requiring 3-4 operations. The distances between locations are provided in Table 1 , with additional details in Table 2 . The effectiveness of NvPSO in minimizing the objective function was assessed by comparing its outcomes with those WaOA. Consequently, equations (41) and (42), which govern the machine layout in the integrated model, are excluded from the non-integrated models formulation.

A comparative analysis of the algorithms revealed that NvPSO consistently outperforms the others in minimizing Makespan. While WaOA also delivers promising results, its effectiveness improves significantly with an increased number of iterations, which requires additional time. Despite being less complex than NvPSO, WaO shows potential, especially in scenarios where computational simplicity is a priority.

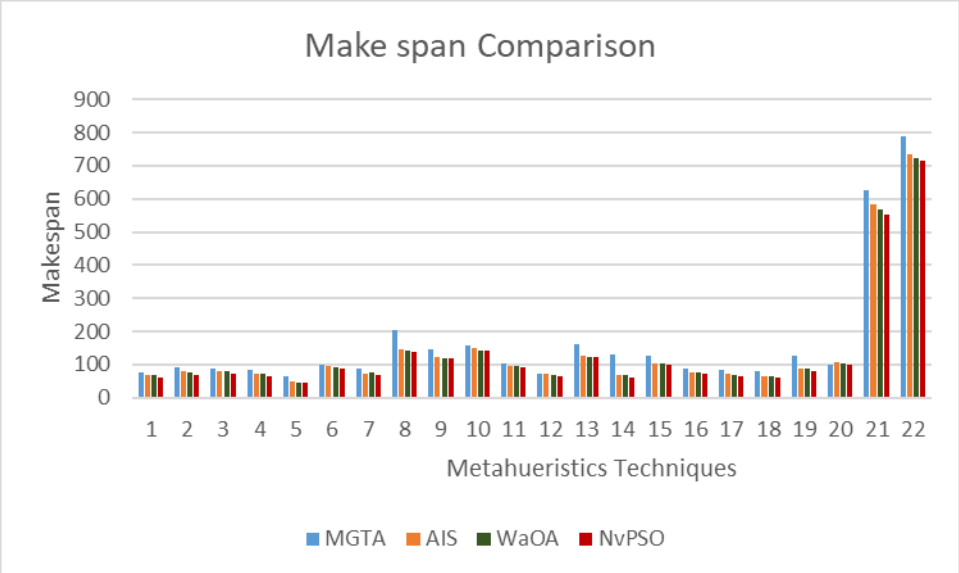


Figure 8: Comparative analysis of an algorithm for 22 Jobsets

However, the difference in Makespan becomes particularly pronounced with larger Jobsets, where NvPSO's superior performance leads to a substantial reduction in overall processing time. This decrease in processing time not only lowers tardiness but also results in significant cost savings.The results showed that NvPSO and WaO satisfy all necessary equations and constraints, confirming its feasibility. This adjustment led to significantly different processing sequences on each machine, as illustrated by the Gantt charts in Figures 9 and 10. The completion times for each job were reduced in the model.

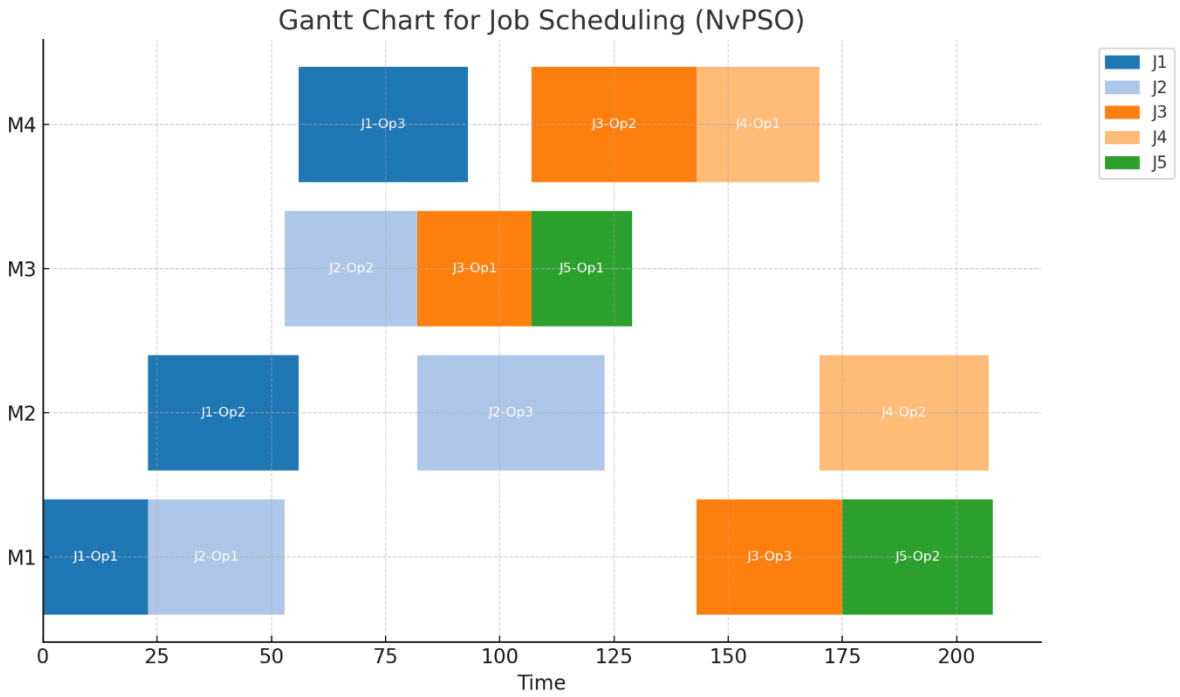


Figure 9: Gantt chart of scheduling using NvPSO

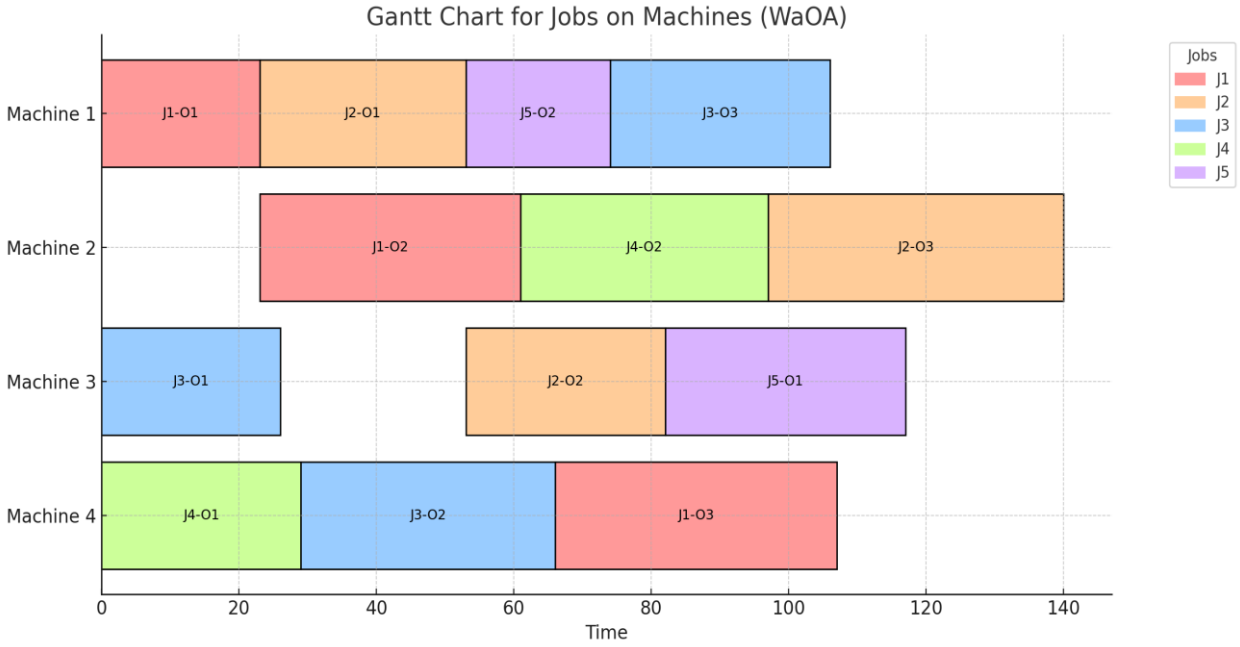


Figure 10: Gantt chart of scheduling using WaOA

By optimizing both scheduling and machine layout, the integrated model provided enhanced flexibility, resulting in a reduction of total energy costs by approximately 9% and a 100% decrease in tardiness penalties.

Table 11: Comparative analysis of the Processing time of both algorithms

	PROCESSING TIME							
	WaOA				NvPSO			
	Machines	OPERATION 1	OPERATION 2	OPERATION 3	**Machines**	OPERATION 1	OPERATION 2	OPERATION 3
JI	**1**	23			**1**	**23**		
	2		38		**2**		**33**	
	3				**3**			
	4			41	**4**			**37**
	Machines	OPERATION 1	OPERATION 2	OPERATION 3	**Machines**	OPERATION 1	OPERATION 2	OPERATION 3
J2	**1**	30			**1**	30		
	2			43	**2**			**41**
	3		29		**3**		29	
	4				**4**			
	Machines	OPERATION 1	OPERATION 2	OPERATION 3	**Machines**	OPERATION 1	OPERATION 2	OPERATION 3
J3	**1**			32	**1**			32
	2				**2**			
	3	26			**3**	**25**		
	4		37		**4**		**36**	
	Machines	OPERATION 1	OPERATION 2	OPERATION 3	**Machines**	OPERATION 1	OPERATION 2	OPERATION 3
J4	**1**				**1**			
	2		36		**2**		37	
	3				**3**			
	4	29			**4**	**27**		
	Machines	OPERATION 1	OPERATION 2	OPERATION 3	**Machines**	OPERATION 1	OPERATION 2	OPERATION 3
J5	**1**		21		**1**		33	
	2				**2**			
	3	35			**3**	**22**		
	4				**4**			

Table 12: Comparative analysis of the Energy minimization

Jobs	WaOA	NvPSO	Process Energy
JI	1530	1395	
J2	1530	1500	
J3	1425	1390	
J4	975	960	
J5	840	825	
Jobs	WaOA	NvPSO	Idle Power
JI	630	550	
J2	623	603	
J3	598	587	
J4	302	301	
J5	297	294	
Machines	Brief Dormant Mode	Extended Dormant	States

		Mode	
1	300	0	
2	305	0	
3	245	0	
4	288	0	

The table provides a detailed breakdown of energy consumption and operational states for different jobs and machines in a production environment. Job J1 consumes 1530 units of energy during active operations and 1395 units during non-value-adding operations. This suggests that while the job is energy-intensive during active processing, it also has significant energy use in supportive operations.Job J5 has the lowest energy consumption, with 840 units during active operations and 825 units during non-value-adding operations, indicating that both operational phases consume nearly the same amount of energy.

1. Energy Inefficiencies in Non-Value-Adding Operations: For many jobs, such as J1, the energy consumed in NvPSO is nearly as high as during active operations (WaOA). This suggests that there may be room for improvement in reducing these non-essential energy costs.

2. Idle Power Consumption: Some machines consume a significant amount of energy even when idle. as seen with the machines handling J1 and J2. Strategies such as implementing better power management or reducing idle times could lead to significant energy savings.

3. Dormant Mode Management: The machines show zero energy consumption in Extended Dormant Mode, indicating effective energy management when machines are not used for extended periods. However, there's still energy use in Brief Dormant Mode, which might be optimized further.

5.4.2 Statistical analysis

The table presents data on various job sets, analysing their energy consumption in different operational states (WaOA and NvPSO) and assessing their performance through additional metrics such as AIS (Average Index Score), Mean, Standard Deviation (St.Dev), and ranking based on deviations from these values. The analysis of the Table 4 is as follows :

Energy Consumption Trends: Job sets with higher AIS values generally have higher energy consumption, as seen with Job Sets 21 and 22. However, their high AIS suggests that these job sets may involve more complex or critical tasks that justify higher energy use.

Efficiency Insights: Job sets with smaller deviations (like Job Set 5) demonstrate stable energy consumption, indicating consistent performance. In contrast, job sets with large negative deviations in Nv-PSO and WaOA (like Job Sets 21 and 22) suggest inefficiencies, particularly in managing non-value-adding operations.

Performance Evaluation: The rank deviations help identify areas where specific job sets might need improvement. Job sets with high negative ranks could benefit from energy-saving strategies, especially in non-value-adding operations.

Table 13: Statistical analysis of both algorithms

Job set	WaOA	NvPSO	AIS	Mean	St.Dev	Rank(+)Nv-PSO	Rank(+)WaOA
1	67	60	69	65.3	2.8	-2.3	-0.6
2	78	69	82	76.3	2.8	-9	-1.2
3	80	71	80	77.0	9.0	-2.3	-0.5
4	71	65	72	69.3	2.8	-0.7	-0.4
5	45	44	48	45.7	0.4	-0.3	-1.3
6	93	87	95	91.7	1.8	-3	-0.6
7	75	68	74	72.3	7.1	-4	-0.4
8	141	140	145	142.0	1.0	-1.5	-1.2
9	117	117	122	118.7	2.8	-1.5	-6
10	143	142	149	144.7	2.8	-0.7	-7
11	94	93	96	94.3	0.1	-0.3	-0.6
12	67	64	71	67.3	0.1	-0.7	-1.2
13	124	123	126	124.3	0.1	-0.3	-0.6
14	67	61	70	66.0	1.0	-2.3	-1.3
15	103	100	104	102.3	0.4	-0.3	-0.4
16	75	72	75	74.0	1.0	-0.3	-0.5
17	70	65	72	69.0	1.0	-0.7	-0.6
18	63	60	64	62.3	0.4	-0.3	-0.4
19	88	79	89	85.3	7.1	-8	-0.4
20	104	99	107	103.3	0.4	-3	-1.3
21	568	553	582	567.7	0.1	-11	-9
22	722	713	733	722.7	0.4	-10	-8

5.5 Efficacy of NvPSO and WaOA on Flexible job shop scheduling

The NvPSO and WaOA have both been shown to be effective in scheduling and reducing energy use at the same time. To further evaluate their effectiveness, we used these algorithms to flexible job shop scheduling (FJSS), a more sophisticated scenario in which jobs are processed on various machines. This extra flexibility complicates the scheduling procedure since it requires optimising several objectives, such as minimising Makespan and energy usage.We evaluated NvPSO and WaOA on well-known benchmark problems in FJSS, which are widely used in the research community to assess scheduling algorithms. The findings revealed that both algorithms could efficiently find high-quality solutions, effectively dealing with the intricacies of FJSS while balancing scheduling efficiency and energy minimisation. This displays their strength and adaptability, demonstrating their suitability for usage in a variety of industrial applications requiring variable job shop scheduling.

Test Instance -Kacem

The data for Test Instance 1 is sourced from Kacem et al. (2002b), comprising five problem instances. Four cases have job-to-machine sizes of KA1: 4 × 5, KA2: 10 × 7, KA3: 10 × 10, and KA4: 15 × 10, all of which exhibit total operation flexibility. The fifth instance, KA5, involves 8 jobs and 8 machines with partial flexibility.

Performance Analysis of WaOA and NVPSO

This section compares the performance of WaOA and NVPSO, as well as other metaheuristics used in the literature to address Kacem problem situations. The table summarises the findings of the comparison between NVPSO and HAdFA. The first column indicates the problem kind, with 'n' jobs and'm' machines. The objectives being reviewed include. NVPSO consistently outperforms in all WaOA situations, notably in large-scale issues, thanks to its adaptive parameter technique, which improves performance over NVPSO. The computational time for both algorithms shows minimal variation;

however, even a few seconds can make a significant difference in real-time complex scenarios.

Figure 11: Graphical Analysis (a)- (e)

Table 14: Comparative analysis of the algorithm

	DE	MOPSO	BEG NSGA	PSO+R RHC	ADCSO	IH PSO	HAdFA	HFPA	WaOA	NvPSO
	MM	MM	MM	MM	MM	MM	MM	MM	MM	MM
KA1	11	11	11		12	11	10	11	10	10
	11	13	11		11		11	12	11	9
		12	12				11		10	10
			13				12		13	10
KA2	11				11		11	11	11	8
									11	11
	11				11		11	11	11	10
	12				12		11	12	10	10
							12		12	11
KA3	8	8	7	7	7	7	7	7	7	6
	7	8	7	7	7	8	7	7	7	7
	7	7	8	8	8		8	8	7	7
		7	8	8			8		7	6
KA4	11	11	11	11	12		11	11	11	11
	12	11	12	14	11		11	11	11	11
	14	16	14	14	16	14	14	14	12	14
KA5	15	14	15	15	15	15	14	15	14	13
		16	16	16	14		16	16	16	16
		15	16	16			16		16	15

The NvPSO algorithm has demonstrated a clear edge over its competitors in terms of performance, as evidenced by its superior results across multiple instances. Specifically, NvPSO consistently achieves the lowest Makespan in four out of five tested scenarios—KA1, KA2, KA3, and KA5. The standout performance is particularly notable in instance KA3, where NvPSO attains an exceptionally low Makespan of 6, the lowest observed among all the algorithms across any of the instances.In contrast, while many of the other algorithms show a similar range of Makespan values, typically between 10 and 16, there are distinct differences in their performance. Most algorithms display a reasonable level of consistency, but certain ones—such as Differential Evolution (DE), Basic Evolutionary Genetics (BEG), and variations of Particle Swarm Optimization (PSO+R) and Randomized Hill Climbing (RHC)—occasionally produce higher Makespans, especially in instances KA4 and KA5. These discrepancies are particularly pronounced in KA4 and KA5, where the Makespan values vary widely. Some algorithms in these instances yield Makespans as high as 16, while others manage to achieve more favorable outcomes in the range of 11 to 13.The broader variation in performance across instances KA4 and KA5 highlights their increased difficulty compared to other instances. The substantial range in Makespan values indicates that these scenarios present a greater challenge to most algorithms, with performance being less consistent and more variable.In summary, NvPSO is distinguished as the best-performing algorithm overall due to its ability to achieve the lowest Makespan in multiple instances. Although other algorithms generally show reasonable consistency, their performance is less reliable, especially in more challenging instances like KA4 and KA5, where they exhibit greater variability in Makespan results.

5.6 Conclusion:

This research provides a comprehensive evaluation of Particle Swarm Optimization (PSO) models, specifically focusing on the novel variant NvPSO, in addressing complex scheduling and energy minimization problems. By utilizing a set of 13 diverse jobsets, we tested the efficacy of NvPSO against traditional search algorithms, including Artificial Immune System (AIS) and Modified Genetic Tabu Algorithm (MGTA), as well as the Walrus Optimization Algorithm (WaOA). The goal was toward

performance of these algorithms in minimising Makespan and energy consumption while providing a robust statistical analysis of their effectiveness.

Key Findings:

1. Superior Performance of NvPSO: Our results consistently demonstrated that NvPSO outperforms WaOA and traditional algorithms in minimizing Makespan and energy consumption. NvPSO achieved significant reductions in processing time, particularly for larger and more complex jobsets. This performance advantage is attributed to NvPSO's adaptive parameter settings, which allow it to better navigate complex scheduling scenarios and optimize both scheduling efficiency and energy use.

2. Energy Efficiency: NvPSO proved to be highly effective in reducing energy consumption. The model led to a drop in total energy costs by approximately 9% besides completely eliminated tardiness penalties. The efficiency of NvPSO in managing both active and non-value-adding operational phases highlighted its potential for substantial energy savings. In contrast, WaOA, while effective, showed improvement in efficiency only with an increased number of iterations, making it a viable alternative where computational simplicity is a priority.

3. Statistical Insights: Comprehensive statistical analysis revealed that NvPSO maintained more consistent performance across different jobsets compared to its counterparts. Jobsets with smaller deviations demonstrated stable energy consumption and effective performance, while those with larger deviations suggested areas for potential improvement. The results underscore NvPSO’s ability to handle diverse scenarios with greater consistency and reliability.

4. Comparison with Traditional Algorithms: NvPSO’s superior performance was evident in comparison to AIS and MGTA, achieving lower Makespan values and better energy management. The comparative analysis highlighted the advantages of advanced PSO models over traditional methods, particularly in complex scheduling tasks. NvPSO’s efficiency in achieving optimal results was further demonstrated by its ability to manage larger jobsets effectively, reducing overall processing time and costs.

5. Challenges and Future Directions: While NvPSO proved to be highly effective, the study also noted that WaOA's performance improved with additional iterations, suggesting a trade-off between computational complexity and effectiveness. Future research should focus on optimizing WaOA or exploring hybrid models that integrate the strengths of NvPSO and WaOA. Additionally, further investigation into real-world applications and large-scale implementations could provide deeper insights into the practical benefits of these algorithms.

Chapter 6

Simultaneous scheduling of machine tools and AGVS using Machine Learning

6.1 Introduction

In today's manufacturing contexts, the integration of new technologies such as automation, data analytics, and machine learning has become critical to improving operational efficiency and flexibility. One of the most critical and complex issues in such environments is scheduling machine tools and Automated Guided Vehicles (AGVs) at the same time. Machine tools are essential to production operations, whereas AGVs are critical enablers for material transfer across several workstations. Traditionally, these two components have been scheduled independently, resulting in suboptimal resource utilisation, higher idle times, and longer production cycles.

With the advent of Industry 4.0, there is an increasing demand for intelligent and adaptive scheduling solutions capable of dynamically responding to real-time changes and uncertainties on the shop floor. Machine learning is a promising approach that uses data-driven methodologies to forecast job priorities, optimise task sequences, and reduce system bottlenecks. Manufacturers can create a more synchronised relationship between machine tools and AGVs by incorporating machine learning models into scheduling algorithms, resulting in increased throughput, shorter lead times, and higher productivity.

This thesis investigates the design and implementation of a machine learning-based framework for scheduling machine tools and AGVs in a flexible manufacturing system (FMS). It seeks to solve the problem's multi-objective nature by adding variables such as work priority, transport time, machine availability, and AGV path optimisation. The proposed approach is intended to provide a scalable and intelligent solution to scheduling issues in smart manufacturing environments.

6.2 Problem statement and methodology

The problem statement pertains to FMS configuration shown in Figure 12 . It consists of 4 CNC m/c's equipped with AGVs, loading-unloading stations and AS/RS. The case study contains 10 task sets, with each set including 4 to 8 job sequences ordered in different orders (Bilge & Ulusoy, 1995). Four different layouts are considered, resulting in a total of 82 problems generated. Within the configuration, there are loading and unloading stations where unfinished jobs are assigned and finished jobs are submitted. Nonetheless, constrained by financial considerations, there exists only a single instance of every tool type. These tools are contained within a central tool magazine (CTM), which caters to numerous machines. Furthermore, an automated storage and retrieval system (AS/RS) is in place to stock ongoing tasks.

Several assumptions are made in this context:

1. The type of m/c remains fixed.
2. The no of m/cs is predetermined.
3. The speed of the Automated Guided Vehicle (AGV) is 40m/min.
4. The distances between the machines are known.
5. Each tool/machine possesses the capacity to manage just a single task concurrently.
6. Pre-emption of operations on the machines is not permissible.
7. The essential machines and tools are recognized before arranging each operation.
8. Every task comprises a pre-established series of operations, each with a distinct sequential arrangement and known processing durations.
9. The processing times include setup times.
10. A single Tool Transfer (TT) device is accountable for moving tools during the FMS, and tool switch times between m/c are not considered.

In summary, the problem statement outlines the configuration of an FMS with specific machines, job sets, and layouts. It highlights the limited availability of tools and their storage in central tool magazines(Raj et al., 2014) Assumptions regarding machine types, numbers, AGV speed, distances, operation pre-emption, and tool transfer are also defined. The travel time data (TTD) and job sets data are stated in Appendix I.

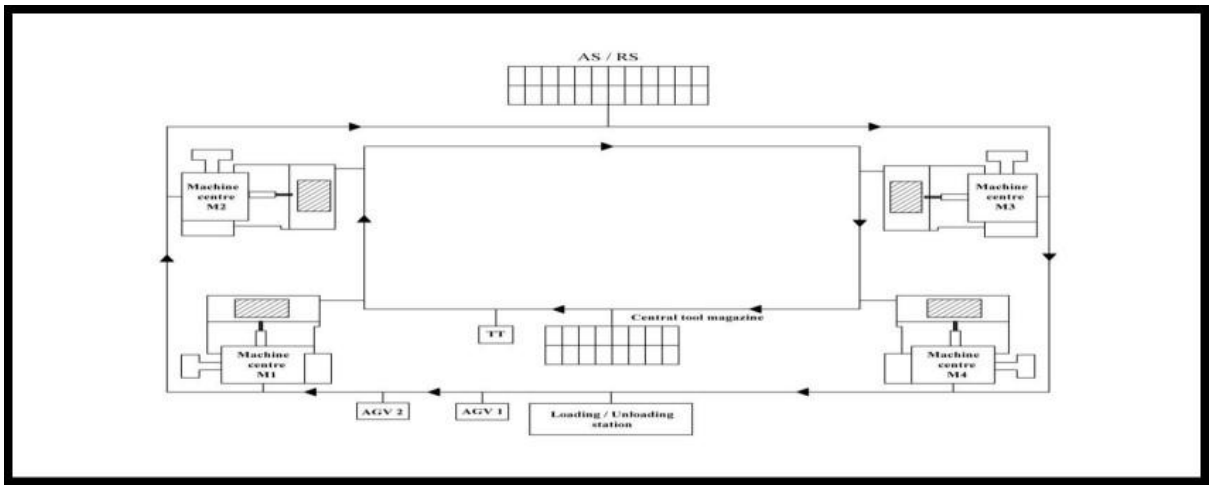


Figure 12: FMS Configuration for machine tools and AGV's

The objective function can be well-defined mathematically as follows:

$$Minimise\ Z = \left(\sum_{i=1}^{n} C_k\right) \quad (52)$$

where,

Operation completion time of k^{th} operation of l^{th} job

$$O_{kl} = T_{kl} + TWT_{kl} + TTT_{kl} + P_{kl} \quad (53)$$

T_{kl} is the entire travel time of the AGVs for k^{th} operation of l^{th} job

$$T_{kl} = ETA_{kl} + LOA_{kl} \quad (54)$$

ETA_{kl} *-Empty trip travel time of AGV.*

LOA_{kl} *- Loaded trip travel time of AGV for* k^{th} *operation of* l^{th} *job.*

TWT_{kl} *is the waiting time of the machine for a requirement of the tool for* l^{th} *operation of* k^{th} *job*

TTT_{kl} *-Tool Transportation Time*

$$TTT_{kl} = ETT_{kl} + LOT_{kl} \quad (55)$$

ETT_{kl} *-Empty trip travel time of Tool Transporter*

LOT_{kl} *- Loaded trip travel time of Tool Transporter*

P_{kl} *is the processing time for* k^{th} *operation of* l^{th} *job*

$$P_{kl} = M_{kl} + S_{kl} + L_{kl} + UL_{kl} + TC_{kl} \quad (56)$$

Job Completion time $C_i = \sum_{k=1}^{l} O\ kl$

where, M_{kl} *= Machining time*

S_{kl} *=Set up time*

L_{kl} *=Loading time*

UL_{kl} *=Unloading time*

TC_k *= Tool changing time*

C_i*= Total Make Span*

6.3 Experimental studies and discussions

In the context of concurrent scheduling for machines and tools, it is frequently assumed that the time required for tool transfers between Central Tool Magazine (CTM), machines, and machines is minimal. However, this assumption becomes impractical when dealing with real-world scheduling circumstances, particularly when tool movements rely solely on tool transporters and transfer times are comparable to production periods. Disregarding tool transfer times makes the scheduling results unsuitable for implementation. This study focuses on tool flows. The goal is to create optimised sequences that reduce the makespan in a multi-machine Flexible Manufacturing System (FMS). The FMS's performance is expected to improve as a result of efficient resource utilisation, effective process integration, and timetable coordination. The FMS setup mentioned in this study is derived from earlier research (Raj et al., 2014).

Table 15 : Job Sets

Job Set	No. of Jobs	Max no. of operations in a job	Total no. of operations	No. of Machines	No. of Tools
1	5	3	13		
2	6	3	15		
3	6	4	16		
4	5	5	19		
5	5	3	13		
6	6	3	18		
7	8	3	19		
8	6	4	20		
9	5	4	17		
10	6	4	21	4	4
11	7	3	18		
12	8	3	19		
13	5	5	19		
14	9	3	21		
15	10	3	22		
16	8	3	19		
17	7	3	17		
18	9	3	20		
19	5	5	19		
20	10	3	24		
21	15	8	110	6	6
22	20	8	151	8	8

A novel hybrid procedure called Dy-PSO has been devised for addressing scheduling challenges involving 22 job sets. The problem matrix utilized for this specific scenario is outlined in Table 14. To implement this innovative strategy, programming was accomplished using MATLAB software, version R2016b or later. MATLAB's capabilities proved instrumental in tackling intricate scheduling problems encompassing diverse job types and multiple machines The proposed hybrid approach yielded diverse optimal schedules for the FMS, prompting a thorough comparison and analysis of their performance.

6.3.1 Parameter Values:

- **Inertia Weight** (w):0.719
- **Cognitive and Social Coefficients** (c_1,c_2): c_1=1.5,c_2=1.
- **Population Size** (N): 80
- **Maximum Iterations** (R): 50
- **Dimensionality (n)**: 6
- **Trials (M)**: 15

6.3.2 The Performance Assessment of Proposed Algorithms Versus Other Heuristics

To gauge efficacy of the algorithm, it was compared against alternative algorithms, like Particle swarm optimisation (PSO) and other algorithms like MWR-Most work remaining, LWR-Least work remaining, LPT-Longest Processing time, SPT- Shortest processing time, MOR-Most operation remaining, LOR-Least operation remaining, ND- MWR- Non-Delay Most work remaining, ND-LWR- Non-Delay Least work remaining, ND- LPT - Non-Delay Longest Processing time, ND-SPT-Non-Delay Shortest processing time, ND- MOR- Non-Delay Most operation remaining, ND-LOR-Non-Delay Least operation remaining, MGTA-Modified Giffer and Thompson Algorithm, AIS-Artificial Immune System, SOS- Symbiotic Organisms Search Algorithm, CSA- Crow Search

Algorithm, FPA-Flower Pollination Algorithm, as documented in various sources. The comparison results showing the percentage of improvement from the best algorithm and worst algorithm is captured in Table 15 and depicted in Figures 2 and 3 spotlighting the algorithm's prowess in addressing 22 job sets.

Table 16 : Minimum Makespan of Simultaneous Scheduling of Machine Without Tool Transfer Time

Job set	MWR	LWR	LPT	SPT	MOR	LOR	ND-MWR	ND-LWR	ND-LPT	ND-SPT	ND-MOR	ND-LOR	MGTA	AIS	PSO	Dy-PSO
1	104	116	77	100	125	116	90	101	86	83	88	73	77	69	72	**67**
2	112	133	111	125	153	133	96	107	98	90	98	87	90	82	82	**79**
3	90	151	148	139	121	141	87	115	108	105	115	105	87	80	81	**77**
4	80	152	119	132	77	154	74	83	89	73	73	78	84	72	75	**72**
5	72	105	78	66	60	87	66	66	75	66	72	75	66	48	49	**46**
6	100	109	100	100	108	140	95	95	95	104	115	101	98	95	95	**92**
7	120	150	107	139	89	127	84	74	101	74	74	84	87	74	78	**73**
8	215	213	211	204	215	213	160	153	153	151	154	165	204	145	143	**142**
9	182	146	184	156	158	158	139	126	160	134	144	130	145	122	125	**119**
10	239	238	244	183	224	217	164	152	182	158	165	164	158	149	150	**149**
11	128	218	153	171	150	153	105	137	109	124	142	101	104	96	**96**	100
12	134	134	134	134	134	134	71	83	77	76	75	82	72	71	72	**68**
13	209	226	195	204	211	236	137	152	139	161	166	136	161	126	128	**127**
14	100	160	121	127	105	132	84	92	82	71	83	84	132	70	69	**66**
15	140	177	162	178	177	184	104	130	119	129	139	106	127	104	104	**103**
16	123	137	96	114	108	123	89	83	90	88	90	80	89	75	75	**71**
17	109	89	75	121	94	99	74	82	76	82	81	74	83	72	75	**70**
18	82	146	98	116	130	151	71	68	88	81	86	70	79	64	68	**63**
19	113	187	148	142	157	187	91	109	109	127	121	104	127	89	90	**83**
20	115	183	163	139	172	160	107	125	115	114	127	112	98	107	110	**107**
21	862	876	755	655	709	623	652	661	661	623	623	644	626	582	590	**581**
22	779	1300	768	757	1264	1255	784	742	781	748	780	804	787	733	748	**721**

The Dy-PSO algorithm was executed across various iterations to pursue optimal results. While initial iterations did not yield much difference but enhancements emerged as the iteration count escalated. The experimentation culminated at 500 iterations, a point where no notable further improvement in the optimal value was observed.

The proposed heuristics are rigorously examined and evaluated across an array of diverse job sets, as delineated in Table 1. The central focus of these examinations is the minimization of the makespan, a pivotal performance metric. The resultant outcomes, comprising a comprehensive dataset of 82 distinct problem instances, spanning various cases and layouts, are meticulously presented and subjected to a meticulous comparative analysis within subsequent sub-sections. Each of these subsections is

thoughtfully structured to correspond to specific cases, thereby enabling a systematic assessment. Of paramount importance in this assessment is the notion of "WTT" AND “WOTT” an acronym encapsulating the concept of Minimum Makespan in the context of Simultaneous Scheduling of Machines “With Tool Transfer Time” and “Without Tool Transfer Time”. This metric provides a succinct representation of the minimal makespan achieved through the devised approach, which seamlessly incorporates both machine and tool schedules while accounting for the intricacies of tool transfer durations. The provided table No.15 clearly indicates that the DY-PSO marked bold method consistently achieves superior makespan results in comparison to the other methods listed in the table. It is evident that DY-PSO stands out and has demonstrated remarkable performance when compared to various methods documented in the existing literature. The empirical evidence substantiates that an impressive 86% of the outcomes from the analysis involving 22 job sets exhibit significantly reduced makespan values when utilizing the DY-PSO approach in contrast to alternative methods.

6.3.4 Performance Comparison of Proposed Algorithms with Worst and Best Algorithms from existing literature

The table provided presents a distinct contrast, highlighting DY-PSO's superior performance in comparison to the conventional PSO. The evaluation involves a comparison with the most and least effective solutions using the available data in the table. Notably, DY-PSO exhibits a significant advantage by achieving a makespan approximately 4.2% lower than the optimal makespan attained by the best historical algorithm. Additionally, the difference in makespan between DY-PSO and the weakest algorithm is approximately 60%. The visual representation graphically depicts the percentage variation from the optimal outcomes yielded by all algorithms, ranging from the most successful to the least effective ones. The presented tabular data delineates the job, machine, and tool sequences produced through Dy-PSO across all 22 job sets, accompanied by the corresponding makespan for each sequence.

Table 17: % Improvement from Best and Worst Algorithm

	% IMPROVEMENT FROM BEST		**% IMPROVEMENT FROM WORST**	
Job set	PSO	Dy-PSO	PSO	Dy-PSO
1	-4.3	2.9	42.4	46.4
2	0.0	3.7	46.4	48.4
3	-1.3	3.8	46.4	49.0
4	-4.2	0.0	51.3	53.2
5	-2.1	4.2	53.3	56.2
6	0.0	3.2	32.1	34.3
7	-5.4	1.4	48.0	51.3
8	1.4	2.1	33.5	34.0

9	-2.5	2.5	32.1	35.3
10	-0.7	0.0	38.5	38.9
11	0.0	-4.2	56.0	54.1
12	-1.4	4.2	46.3	49.3
13	-1.6	-0.8	45.8	46.2
14	1.4	5.7	56.9	58.8
15	0.0	1.0	43.5	44.0
16	0.0	5.3	45.3	48.2
17	-4.2	2.8	38.0	42.1
18	-6.3	1.6	55.0	58.3
19	-1.1	6.7	51.9	55.6
20	-2.8	0.0	39.9	41.5
21	-1.4	0.2	32.6	33.7
22	-2.0	0.0	42.5	40.5

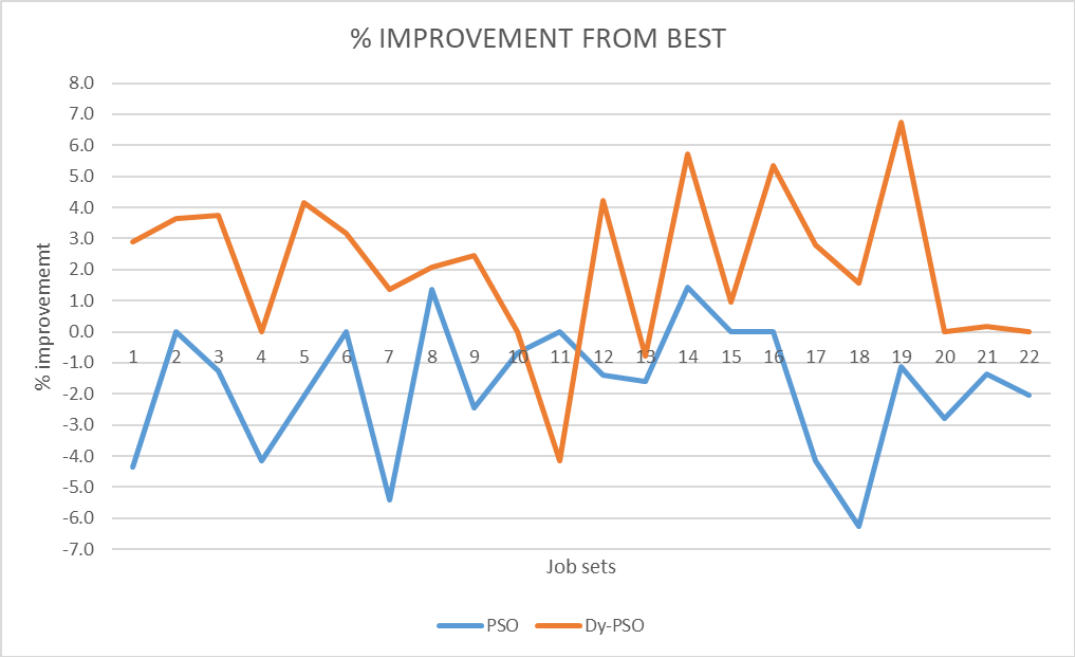


Figure 13: %Improvement from worst

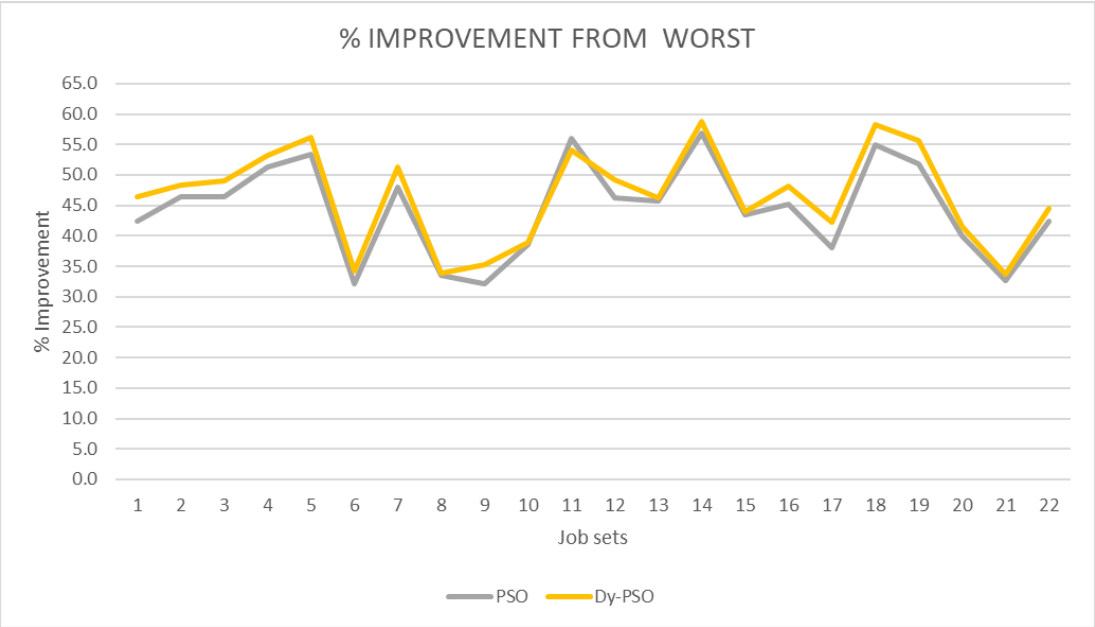


Figure 14: %Improvement from best

6.3.5 Job sequences of 22 job sets

The provided table presents various schedules for 22 job sets, along with their respective make spans. In this context, a sequence like "3244" indicates that job 3 underwent operation 2 using machine 4 and tool 4. The objective is to create an efficient schedule that optimally utilizes all available machines and tools, resulting in the minimum total makespan for all 22 job sets.

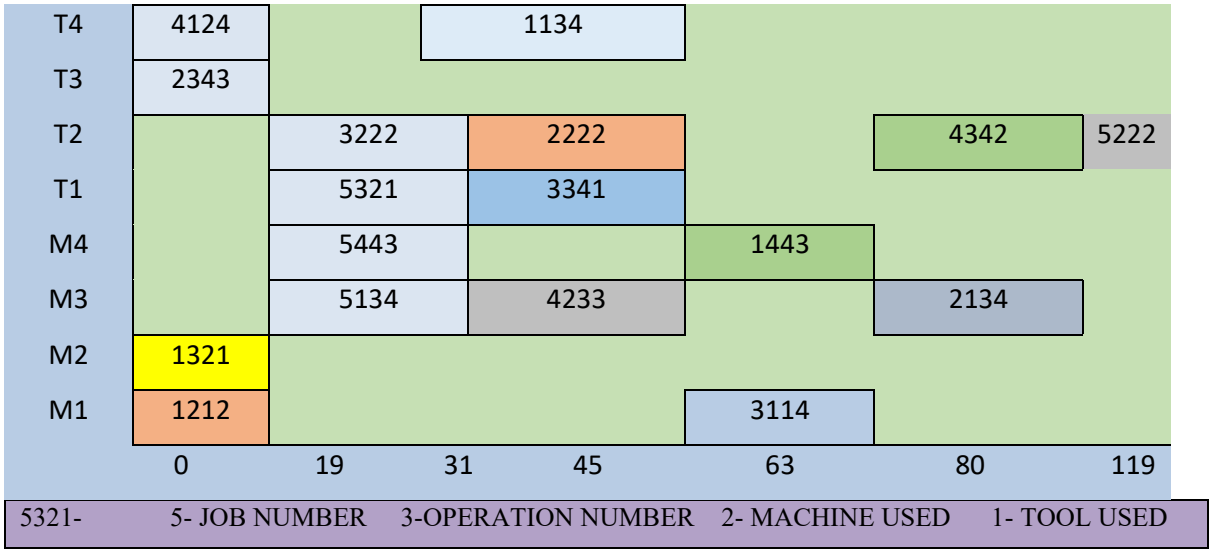
Table 18: Scheduling of 22 Job sets

Sr. No	***Job Sets***	***SEQUENCES***	***Make span***
1	1	[(3244)(5132)(1224)(3312)(1341)(4143)(2112)(5211)(1113)(3131)(2321)(2233)(4224)]	67
2	2	[(3223)(1241)(6114)(2122)(4123)(6333)(4342)(2244)(4231)(3114)(1113)(5114)(5221)(5343)(3232)]	79
3	3	[(3113)(5333)(1232)(5222)(4242)(2243)(6123)(6342)(5444)(4133)(6414)(5114)(2121)(5222)(1114)(3221)]	77
4	4	[(5223)(1142)(4422)(3122)(5422)(3234)(1312)(5344)(2131)(5111)(3432)(4311)(1213)(2344)(5533)(3313)(4243)(2222)(1321)(4122)]	72
5	5	[(3312)(5211)(4142)(5132)(2112)(3244)(2323)(1344)(1221)(2231)(1114)(3133)(4223)]	46
6	6	[(6111)(1344)(4344)(5344)(3233)(2344)(3122)(6344)(5233)(4122)(2222)(1222)(3344)(4233)(2111)(1111)(5344)(5111)]	92
7	7	[(2122)(3244)(5111)(8343)(5223)(7331)(1244)(2243)(4133)(5233)(6121)(3122)(7113)(7224)(4244)(6343)(8114)]	73
8	8	[(2233)(1232)(6334)(5221)(1121)(4343)(5332)(5114)(6344)(4121)(1343)(6112)(3233)(2124)(4232)(5443)(6441)(3122)(2342)(6223)]	142
9	9	[(5321)(4124)(5443)(1134)(1212)(5134)(2343)(3341)(1321)(4233)(3222)(1443)(2222)(2134)(31	119

		14)(4342)(5222)]	
10	10	[(2342)(5222)(1233)(3313)(5331)(2123)(6134)(4123)(4344)(1322)(3134)(1441)(3232)(4213)(3222)(3114)(6331)(1114)(4232)(5233)(6442)]	149
11	11	[(2132)(1222)(1334)(5124)(7243)(7322)(3124)(6114)(3322)(1114)(3244)(5241)(4232)(5311)(4114)(6244)(2223)(7133)]	100
12	12	[(4243)(2123)(6341)(7124)(5221)(1134)(3142)(8214)(4122)(1242)(3133)(7231)(7323)(6142)(5113)(8342)(7241)(2212)(8133)]	68
13	13	[(4133)(3113)(4344)(5533)(2123)(4411)(5332)(4212)(2314)(5243)(2242)(4522)(3422)(3231)(1243)(5111)(5424)(1114)(3344)]	127
14	14	[(6234)(4121)(9211)(2223)(8134)(5214)(2111)(4332)(8313)(7113)(1142)(3134)(2233)(3212)(4243)(9121)(6142)(1222)(5124)(8224)(2344)]	66
15	15	[(2231)(6132)(4114)(3133)(9133)(9241)(5242)(1312)(10 142)(3224)(1142)(8233)(5133)(10 233)(1222)(7123)(2113)(2344)(6114)(4223)(8214)]	103
16	16	[(1244)(3244)(2122)(7331)(5111)(5233)(1114)(8222)(4244)(4133)(8114)(6232)(8343)(6343)(6121)(2243)(7224)(3122)]	71
17	17	[(5211)(1213)(4344)(6231)(7231)(6323)(2343)(2121)(3243)(4111)(5132)(6121)(3122)(4223)(1143)(2222)(7114)]	70
18	18	[(9242)(6113)(2133)(8232)(3142)(1113)(4214)(7223)(5132)(8332)(3232)(9131)(1223)(6241)(8114)(7131)(4131)(5243)(9141)(2241)]	63
19	19	[(5211)(3231)(2314)(2123)(1243)(1114)(5424)(3113)(3344)(4133)(4344)(4411)(2242)(5243)(5533)(5332)(3422)(4521)]	83
20	20	[(3242)(7131)(9112)(10 141)(5323)(1213)(5223)(6132)(7323)(9234)(8231)(4343)(4133)(2223)(6322)(3114)(4242)(10 222)(1143)(5343)(5123)(7233)(2131)]	107
21	21	[(9161)(12 122)(12 241)(12 353)(2132)(2246)(6132)(9252)(6241)(12 424)(9343)(12 535)(2354)(6356)(9434)(12 661)(14 154)(14 225)(1431)(2463)(2512)(2623)(2746)(5122)(6465)(6514)(8161)(8243)(8325)(8412)(8534)(8652)(14 452)(14 511)(14 632)(14 744)(9525)(9616)(9753)(9865)(13 112)(15 166)(8721)(8833)(3111)(3232)(15 222)(15 334)(15 443)(3353)(3424)(3545)(3666)(5233)(12 712)(15 554)(15 661)(15 726)(15 815) (1154)(1223)(1365)(1431)(1546)(1615)(3753)(3824)(4163)(4244)(4325)(4416)(4531)(5344)(6623)(6745) (6824)(10 116)(10 225)(10 334)(10 443)(10 552)(10 661)(13 263)(13 352)(13 431)(13 512)(13 645)(13 726) (13 834)(4652)(5455)(7113)(7235)(7351)(7422)(7544)(7661)(7722)(5566)(5611)(5753)(5812)(11 131)(11 264)(11 325)(11 443)(11 533)(1611)(11 753)(1611)(11 753)(11 812)]	581

22	22	[(6128)(17 171)(6246)(19 122)(2181)(10 177)(15 124)(19 244)(19 377)(10 288)(6364)(4116)(10 311)(11 132) (15 246)(16 112)(17 252)(20 112)(4228)(10 444)(16 234)(10 522)(10 633)(10 733)(1118)(2213)(6428)(10 855)(11 283)(18 128)(19 488)(11 372)(2345)(18 247)(19 565)(18 336)(1247)(3133)(12 128)(13 114)(13 214) (15 368)(18 425)(20 234)(18 534)(18 643)(18 762)(18 881)(15 482)(15 571)(1386)(1421)(6546)(7185) (11 453)(12 271)(13 326)(15 635)(17 313)(.19 656)(12 333)(12 467)(12 542)(7275)(12 678)(12 755)(15 757) (19 722)(12 816)(2467)(3222)(7368)(8152)(11 513)(13 443)(16 381)(17 434)(20 323)(17 556)(8234)(8346) (11 657)(17 626)(8428)(17 767)(8511)(8647)(8783)(8872)(7411)(11 741)(7558)(7627)(7744)(1563)(2538) (3311)(6617)(9165)(13 478)(14 143)(16 478)(20 468)(1654)(2666)(2772)(20 581)(2821)(3477)(6746)(14 223)(16 567)(3544)(14 318)(16 645)(20 673)(3688)(3733)(14 487)(20 715)(14 577)(14 621)(14 765)(16 756) (16 823)(14 826)(20 846)(6873)(9234)(9316)(9436)(9528)(9666)(9787)(4353)(5173)(13 582)(4464)(13 613) (13 765)(4553)(4642)(5285)(5352)(5417)(5531)(5652)(5774)(4731)(4836)]	721

Table 19: Gantt Chart of job set 9



6.3.6 Performance Analysis of Proposed Algorithms with and Without Tool Travel Time

The presented tables offer insights into the make span values of four distinct layouts, factoring in both transfer times and their absence of tools. A relative analysis is conducted between the proposed PSO and Dy-PSO algorithms and the results are shown in Table 19-22. The data displayed unambiguously indicates that across Layouts I, II, III, and IV, when considering tool transfer times, Dy-PSO consistently achieves the minimum make span in contrast to PSO. This pattern holds true even in cases where tool transfer times are not taken into account.

The remarkable outcomes achieved by Dy-PSO can be qualified to its effective combination of exploration and exploitation characteristics. The algorithm incorporates strategies such as identifying and updating stagnant population segments and employing spatial excursion tactics. These strategies prevent the premature convergence of the population and stand as effective methods for enhancing results in complex simultaneous scheduling problems

Table 20: Comparative analysis of PSO and DYPSO with and without tool travel time from Layout 1

LAYOUT -1	PSO		DYPSO	
Job set No.	WTT	WOTT	WTT	WOTT
1	115	89	104	**78**
2	101	71	94	**69**
3	123	100	103	**95**
4	112	97	102	**91**
5	134	119	123	**107**
6	105	87	90	**81**
7	131	105	115	**99**
8	152	123	139	**106**
9	172	157	156	**145**
10	180	163	176	**160**

Table 21: Comparative analysis of PSO and DYPSO with and without tool travel time from Layout 2

LAYOUT -2	PSO		DYPSO	
Job set No.	WTT	WOTT	WTT	WOTT
1	120	99	125	**103**
2	156	132	143	**126**
3	103	87	103	**89**
4	142	123	160	**133**
5	175	161	103	**88**
6	130	125	125	**98**
7	167	151	130	**104**
8	98	80	92	**75**
9	113	100	112	**86**
10	196	175	178	**158**

Table 22: Comparative analysis of PSO and DYPSO with and without tool travel time from Layout 3

LAYOUT -3	PSO		DYPSO	
Job set No.	WTT	WOTT	WTT	WOTT
1	123	98.0	109	**87**
2	148	121.0	127	**98**
3	110	92.0	101	**83**
4	162	147.0	156	**119**

5	134	115.0	127	**109**
6	106	89.0	101	**73**
7	175	157.0	120	**95**
8	99	81.0	91	**70**
9	157	138.0	167	**134**
10	185	167.0	171	**153**

Table 23: Comparative analysis of PSO and DYPSO with and without tool travel time from Layout 4

LAYOUT -4	PSO		DYPSO	
Job set No.	WTT	WOTT	WTT	WOTT
1	136	102.0	112	**81**
2	108	85.0	134	**100**
3	157	131.0	156	**131**
4	122	98.0	103	**83**
5	178	143.0	122	**101**
6	103	83.0	167	**145**
7	149	117.0	135	**121**
8	165	127.0	121	**97**
9	119	91.0	109	**83**
10	142	98.0	105	**85**

6.4 Convergence analysis

To effectively demonstrate the performance advantages of Dy-PSO over other algorithms a convergence rate curve is presented in Figure 13 focusing on the Job sets 20,21 and 22. The X-axis represents the techniques, while the Y-axis indicates the make span value PSO yielded a make span of 110,590 and 748 whereas Dy-PSO achieved a superior make span 107,581 and 721. Figure 13 vividly illustrates Dy-PSO rapid convergence within a brief timeframe compared to other algorithms, leading to a lower make span. This trend holds true across various test problems, showcasing Dy-PSO faster convergence. Nevertheless, the attainment of the best possible make span more than compensates for the slightly extended computation time.

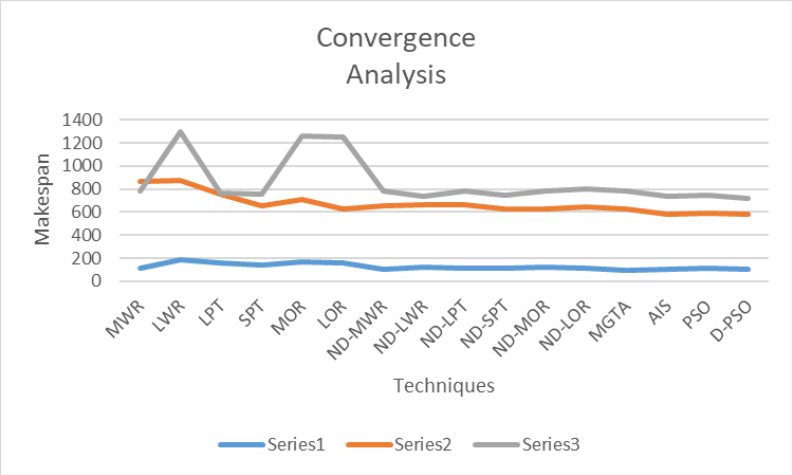


Figure 15: Convergence Analysis

6.5 Hybrid Optimization Approach: Synergizing Genetic Algorithms (GAs) and Machine Learning (ML) for Scheduling Problems

The Scheduling Problem is a critical optimization challenge across industries, requiring the efficient allocation of jobs to machines and tools to minimize overall processing time. This research presents a hybrid approach that integrates Genetic Algorithms (GAs) and Machine Learning (ML) to address complex scheduling problems. Specifically, it focuses on optimizing job sequences while accommodating machine and tool constraints.

The RFR-GA framework integrates multiple components to optimize scheduling problems effectively. First, the Random Forest Regression (RFR) model utilizes features representing scheduling plans, such as machine allocation, job sequence, AGV assignments, and material handling parameters. These inputs are derived from candidate solutions generated by Dynamic-PSO during each iteration. The RFR model is trained on historical scheduling data to predict key performance metrics, including makespan, resource utilization, and flow time. To enhance predictive accuracy, Genetic Algorithms (GAs) are employed to optimize the RFR hyperparameters, such as the number and depth of trees. The RFR model then acts as a surrogate evaluation function for Dynamic-PSO, replacing computationally expensive simulations with efficient predictions. This allows promising solutions to be prioritized for further refinement, while weaker solutions are adjusted or discarded to maintain diversity. Finally, the hybrid approach enables Dynamic-PSO to focus computational resources on fine-tuning high-potential solutions. With real-time feedback from the RFR model, the scheduling plan evolves dynamically, achieving optimized objectives like reduced makespan and improved resource utilization.

The hybrid approach improves computational efficiency by using RFR predictions to reduce reliance on costly simulations, accelerating the search process. It enhances solution quality as GAs optimize RFR hyperparameters for accurate predictions, effectively guiding Dynamic-PSO. Additionally, iterative feedback loops enable dynamic refinement of scheduling plans, achieving superior performance metrics such as reduced makespan and better resource utilization.

6.5.1 Representation of Data

A pandas data frame is employed to represent job scheduling data, encompassing details about each job, such as assigned machine, tool, and processing time. Categorical variables like "Machine" and "Tool" are encoded using Label Encoding for compatibility with ML models.

6.5.2 Machine Learning Model

A Random Forest Regressor serves as the core ML model, trained to predict job processing times based on assigned machines and tools. The objective function of ML model is to minimize Mean Squared Error (MSE) between predicted and actual processing times. The dataset is split into training and testing sets, with ML model trained on the training set using features (machine & tool) and target variable (processing time). The model's hyperparameters are optimized through a Genetic Algorithm, as detailed later.

6.5.2.1 Genetic Algorithm

The Genetic Algorithm uses ordered crossover and shuffle mutation operators to evolve a population over a set number of generations via tournament selection. Parameters like population size, mutation probability, and crossover probability are adjusted to strike a balance between exploration and exploitation. Our study aims to minimise the Mean Squared Error (MSE) between projected and actual processing durations for a specific work sequence.

Let $\mathcal{J}$ represent the set of jobs, and $f:\mathcal{J}\rightarrow\mathbb{R}$ denote the objective function:

f(job sequence) = MSE

We introduce binary decision variables X_i to designate whether job i is scheduled in a particular position in the sequence:

X_i = {

1 if job i is scheduled

0 otherwise

}

The following constraint ensures that each job is scheduled exactly once:

$\sum_{i\in\mathcal{J}} X_i = 1, \forall j\in\mathcal{J}$

The genetic algorithm is employed to find a binary job sequence X that minimizes the objective function through genetic algorithm operations, including crossover, mutation, and selection. The algorithm parameters are set as follows:

pop_size: Population size (number of individuals).

gen_count: Number of generations.

6.5.2.2 Optimization Approach

ML model predictions are integrated into a Genetic Algorithm to optimize job sequences. The Genetic Algorithm employs ordered crossover and shuffle mutation operators, developing a population of potential job sequences over multiple generations.

The proposed Genetic Algorithm's performance is compared with existing scheduling methods to demonstrate its effectiveness in minimizing makespan.

6.5.3 Experimental Results

Experiments on a pre-processed dataset of Job Set-I and Job Set-II evaluate the model and algorithm using metrics like makespan reduction and computational time. Results highlight the superiority of the proposed approach in optimizing job sequences.

Job Set-1

```
Best Job Sequence: [5, 6, 8, 7, 3, 2, 1, 0, 4]
Mean Squared Error (MSE): 11.804983140900571
Mean Absolute Error (MAE): 2.8754632936507925
R^2 Score: 0.1679307037250699
Predicted Processing Time: 56.63035317460319

Final Model Performance:
Mean Squared Error (MSE): 13.181480633786853
Mean Absolute Error (MAE): 3.1265714285714283
R^2 Score: 0.07090885400621305

Actual Processing Time: 57
```

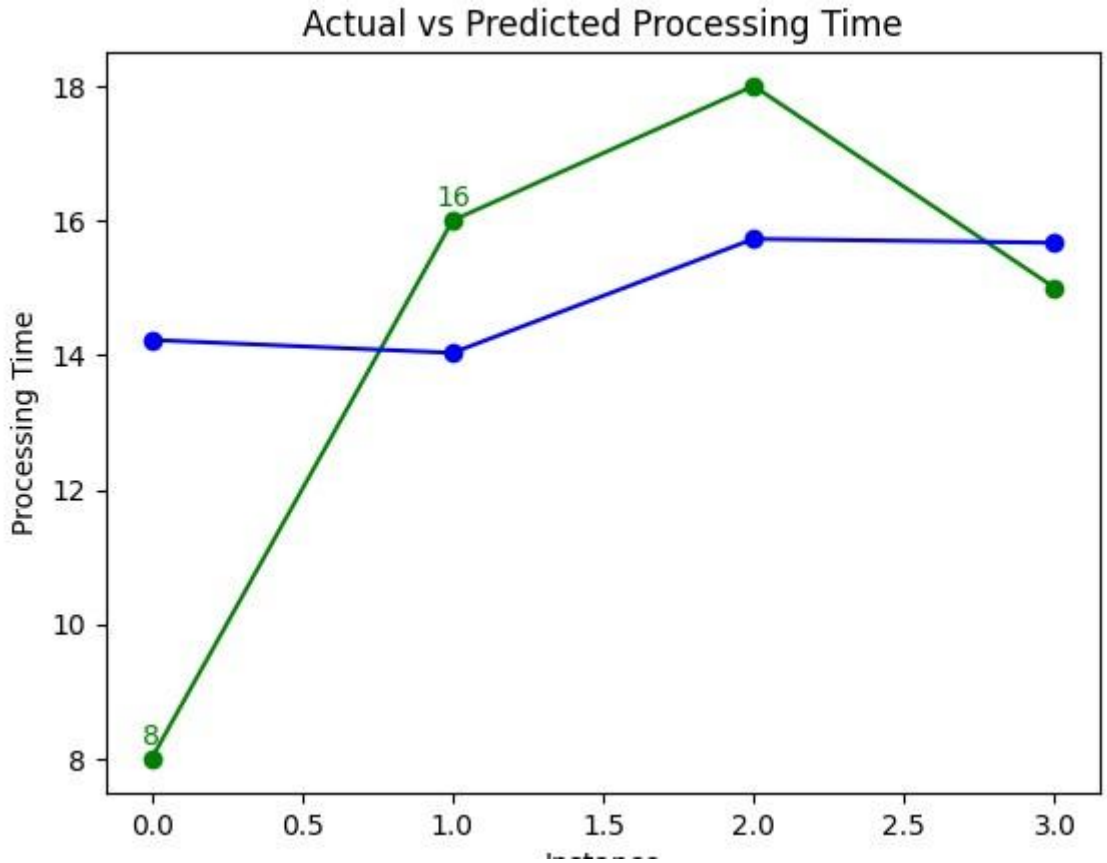


Figure 16: Comparative analysis of actual vs predicted processing time

Job Set-2

Mean Squared Error (MSE): 13.191659999999999

Mean Absolute Error (MAE): 2.558

R^2 Score: 0.07361938202247198

Predicted Processing Time: 58.550000000000004

Final Model Performance:

Mean Squared Error (MSE): 13.191659999999999

Mean Absolute Error (MAE): 2.558

R^2 Score: 0.07361938202247198

Actual Processing Time: 67

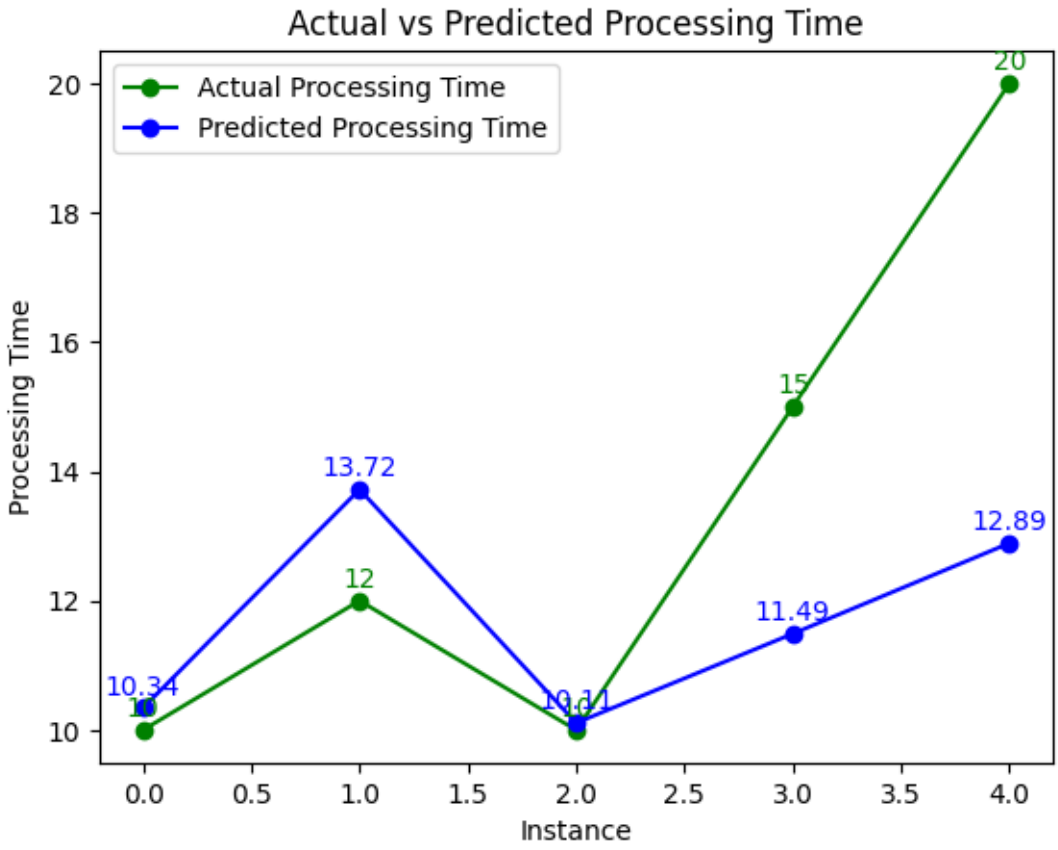


Figure 17: Comparative analysis of actual vs predicted processing time

In the context of using a Random Forest Regressor method in machine learning, the metrics provided are indicators of how well this model is performing in predicting the processing time based on the given features or input variables. Figures 14 and 15 show the comparative status of predicted and actual processing time for Job Set-1 and Job Set-2. This model clearly shows that it gives positive results for small-scale problems (Job Set-1) as compared to medium-scale problems (Job set-2). In Job Set -2 following observations are made as compared to Job Set 1.

- Mean Squared Error (MSE): This is a measure of the average squared difference between the actual processing times & the processing times forecast by model. In this case, the MSE is 13.191659999999999, which means, on average, the squared difference between predicted processing times and actual processing times is approximately 13.192.
- Mean Absolute Error (MAE): Like MSE, MAE calculates the average difference between actual and predicted processing times. The MAE of 2.558 indicates that the model's predictions are typically wrong by about 2.558 units of time.

- R^2 score: R-squared, also known as the coefficient of determination, is a statistical metric that indicates how much of the variance in the dependent variable (processing time) can be explained by the independent variables (features). The R^2 score of 0.07361938202247198 indicates that this model explains approximately 7.36% of the variance in processing time. This value represents the model's goodness of fit; the closer to one, the better.
- Predicted Processing Time: This is the processing time that this model has predicted for a given set of features. In this case, it's approximately 58.55 units of time.
- Actual Processing Time: This is real processing time observed or recorded in given dataset. In this case, it's 67 units of time.

6.5.3.1 Interpreting these metrics collectively:

The MSE and MAE provide information on the accuracy of model's predictions. Lower values are desirable for both metrics, as they indicate smaller errors between predicted and actual values.

R^2 score indicates how well the model fits the data. A higher R^2 value (closer to 1) suggests that the independent variables account for a greater share of the dependent variable's variance.

By comparing the predicted processing time (58.55) with the actual processing time (67), this model's prediction is somewhat off, as indicated by the MSE and MAE.

The research combines Genetic Algorithms and Machine Learning to address the Job Scheduling Problem, resulting in a significant reduction in make span. This hybrid approach contributes to operational efficiency in diverse industries. Future research could explore advanced ML models and further fine-tuning of Genetic Algorithm parameters. Real-world implementations and scalability studies are suggested for practical deployment.

6.6 Statistical Analysis

6.6.1 Friedman Statistical Analysis:

The application of the Friedman statistical analysis technique holds significance as a prevalent statistical approach for evaluating various algorithms. This method permits for a comprehensive assessment of algorithmic performance, with better-performing methods exhibiting lower rankings. In our research, we employ the Friedman statistical analysis to succinctly capture all compared algorithms' accuracy metrics, as shown in Table 23.

Upon examination of Table 23, a clear pattern emerges. Among the considered algorithms, Dy-PSO emerges as the top performer in job set 22,19,16 and 14, closely followed by AIS and PSO. Notably, when evaluating the overall performance, Dy-PSO consistently emerges as the premier choice, trailed by AIS and PSO.

6.6.2 Wilcoxon signed-rank test

A comprehensive assessment was conducted across all 22 problem sets using nonparametric testing to further scrutinize efficacy of proposed algorithms. Among these tests, the Wilcoxon signed rank test, a robust non-parametric method, was employed to discern if there existed notable variations in the

performance of PSO and Dy-PSO. A significance level of $\alpha = 0.05$ was employed, ensuring thoroughness of the analysis. The findings of this rigorous examination revealed that Dy-PSO exhibited significantly superior performance for job set 22 and 19. This discernment underscores the conclusiveness of the proposed Dy-PSO performance in comparison to PSO. A noteworthy aspect observed was the commendable efficiency of Dy-PSO, particularly in addressing larger-scale instances. In contrast, PSO demonstrated its strengths when dealing with smaller-scale instances. This distinction emphasizes the algorithmic prowess exhibited by each approach, with Dy-PSO excelling in managing complex, extensive scenarios, while PSO proved to be well-suited for more modest-sized challenges.

Table 24: Wilcoxon Ranking in PSO and Dy-PSO

Job set	PSO	Dy-PSO	AIS	DIFF-Dy-PSO	ABS-DIFF	DIFF-PSO	ABS-DIFF	Rank(+)Dy-PSO	Rank(+)PSO	Rank(-)Dy-PSO	Rank(-)PSO
1	72	67	69	2	2	0	0	0.6			
2	82	79	82	3	3	0	0	1.3			
3	81	77	80	3	3	-1	1	0.5	0.2		-0.2
4	75	72	72	0	0	-3	3		0.7		-0.7
5	49	46	48	2	2	-1	1	0.6	0.2		-0.2
6	95	92	95	3	3	0	0	0.5			
7	78	73	74	1	1	-4	4	0.2	2		-2
8	143	142	145	3	3	2	2	0.5	0.5		
9	125	119	122	3	3	-3	3	0.5	0.7		-0.7
10	150	149	149	0	0	-1	1		0.2		
11	96	100	96	-4	4	0	0	1.3		-1.3	
12	72	68	71	3	3	-1	1	0.5	0.2		
13	128	127	126	-1	1	-2	2	0.2	0.5	-0.2	-0.5
14	69	66	70	4	4	1	1	1.3	0.2		
15	104	103	104	1	1	0	0	0.2			
16	75	71	75	4	4	0	0	1.3			
17	75	70	72	2	2	-3	3	0.6	0.7		-0.7
18	68	63	64	1	1	-4	4	0.2	2		-2
19	90	83	89	6	6	-1	1	5	0.2		-0.2
20	110	107	107	0	0	-3	3		0.7		-0.7
21	590	581	582	1	1	-8	8	0.2	6		-6
22	748	721	733	12	12	-15	15	6	7		-7

6.7 Summary -

6.7.1 Background and Motivation:

Traditional scheduling approaches typically focus on the allocation of workpieces to machines and tools, treating logistics tasks like material handling and delivery as separate processes. However, in real-world manufacturing environments, efficient scheduling requires the simultaneous consideration of material handling, including Automated Guided Vehicle (AGV) delivery, as these are integral to overall system performance. Material handling influences machine utilization, workflow continuity, and ultimately the makespan. Ignoring such logistics tasks can lead to suboptimal solutions and reduced production efficiency. Addressing these challenges calls for a dynamic scheduling model that integrates logistics with production tasks while ensuring computational efficiency.

6.7.2 Proposed Solution:

To tackle this problem, this study recommends a novel **dynamic Particle Swarm Optimization (dynamic-PSO)** algorithm for Simultaneous Scheduling Problems. The algorithm emphasises minimising the makespan by dynamically adjusting parameters during iterations to enhance solution precision and adaptability. Key improvements include strategies for updating stagnant populations and employing spatial excursion techniques, which collectively enhance the algorithm’s performance in complex, multi-resource scheduling scenarios.

6.7.3 Verification and Results:

The algorithm's efficacy was validated using 22 benchmark problems, demonstrating its ability to generate new Pareto optimal solutions and achieve novel makespan values. Unlike conventional methods, dynamic-PSO effectively integrates material handling constraints and optimises both production scheduling and AGV delivery tasks within a unified framework. Results showed that dynamic-PSO consistently outperformed standard PSO in finding superior solutions while maintaining comparable computational times. Statistical analysis using the Wilcoxon test and comparisons with other metaheuristic techniques (AIS, MGTA, and Fuzzy) highlighted the superiority of dynamic-PSO.The real-world applicability of dynamic-PSO was further verified through its implementation in a complex Job Shop Scheduling Problem, showcasing its capability to handle multi-objective fitness functions involving production and logistics integration.

Chapter 7

Case Study – Application of the Proposed Algorithm in the Indian Manufacturing Industry

7.1 Introduction

To demonstrate the practical relevance of the proposed scheduling and optimisation algorithm, a case study is presented on an Indian manufacturing industry. The focus is on a typical MSME-based automotive component supplier operating in a job-shop environment, where multiple products are manufactured in small batches under tight delivery schedules. Such units often face challenges of machine idle time, order tardiness, bottlenecks, and high energy consumption, making them an ideal candidate for evaluating the proposed approach.

7.2 Case 1: Small Instance (6 Jobs × 4 Machines × 2 AGVs)

The first case study considers a small-scale production scenario representative of automotive component MSMEs located in Ghaziabad (Kasuya GPP). The system involves six jobs processed across four machines, supported by two Automated Guided Vehicles (AGVs) for material handling. Typical products include stiffener brackets, which require multiple machining operations such as turning, milling, drilling, and grinding. This setting reflects the operational reality of many Tier-2 and Tier-3 automotive suppliers where job-shop style manufacturing is prevalent, order sizes are small, and on-time delivery is critical. By applying the proposed algorithm to this instance, the objective is to evaluate its effectiveness in reducing makespan while accounting for machine scheduling and AGV coordination in a realistic MSME environment.

Table 25: Job Set 6 Jobs × 4 Machines × 2 AGVs

Job	Operation Sequence (Machines)	Processing Times (mins)	AGV Travel Times (mins)
J1	M1 → M3 → M4	12, 18, 15	2, 3
J2	M2 → M3 → M1	14, 16, 10	3, 2
J3	M4 → M2 → M3	20, 12, 17	2, 4
J4	M3 → M1 → M2 → M4	15, 10, 18, 14	2, 3, 2
J5	M2 → M4	13, 19	3
J6	M1 → M2 → M3	11, 15, 16	2, 3

7.3 Case 2: Medium Instance (10 Jobs × 6 Machines × 3 AGVs)

The second case study is based on Ghaziabad Precision Products (GPP), a representative medium-scale industry engaged in manufacturing precision-engineered components for automotive, defence, and engineering applications. Typical products include precision shafts, engine components, and roller bearings, all requiring high dimensional accuracy and multiple machining operations such as turning, milling, drilling, grinding, and heat treatment. Operating in a job-shop style system, such industries often

face challenges of 20–25% machine idle time, sequencing delays, and tardiness penalties, particularly for export orders. To address these issues, the proposed algorithm is applied to a production setup with 10 jobs across 6 machines and 3 Automated Guided Vehicles (AGVs), optimizing machine scheduling and AGV routing to reduce makespan, balance workloads, and ensure timely deliveries. This highlights the potential of dynamic optimization for improving efficiency and competitiveness in medium-scale Indian industries like those in Ghaziabad

Table 26: Job set 10 Jobs × 6 Machines × 3 AGVs)

Job	Operation Sequence (Machines)	Processing Times (mins)	AGV Travel Times (mins)
J1	M1 → M2 → M6	18, 22, 17	3, 4
J2	M3 → M5 → M4	20, 25, 21	2, 3
J3	M2 → M6	19, 24	3
J4	M4 → M1 → M3	16, 19, 23	2, 4
J5	M5 → M2 → M6 → M4	22, 21, 20, 23	3, 2, 3
J6	M6 → M1	25, 18	4
J7	M2 → M3 → M5	21, 22, 20	3, 2
J8	M4 → M6 → M2	19, 24, 22	3, 3
J9	M3 → M5	20, 23	2
J10	M1 → M4 → M6	18, 20, 25	3, 4

7.4 Case 3: Large Instance (20 Jobs × 8 Machines × 4 AGVs)

The third case study focuses on heavy engineering and steel fabrication industries located in the Delhi NCR region, which cater to sectors such as oil & gas, infrastructure, and power generation, including supply chains linked with Indian Oil Corporation (IOCL) refineries and projects. Typical products include steel plates, structural beams, and pressure vessel parts, requiring large-scale operations such as cutting, rolling, welding, machining, and heat treatment. These industries face challenges of complex job scheduling, high setup times, bottlenecks in heavy-duty machines, and significant material handling delays, which directly impact project timelines and costs. In this case, a production system with 20 jobs across 8 machines and 4 Automated Guided Vehicles (AGVs) is considered, where the proposed algorithm is applied to integrate machine scheduling with AGV coordination. The objective is to minimize makespan, enhance throughput, and optimize resource utilization in a large-scale industrial environment, thereby demonstrating the scalability and applicability of the approach to the Delhi NCR heavy engineering and refinery-support sector.

Table 27: Job set 20 Jobs × 8 Machines × 4 AGVs

Job	Operation Sequence (Machines)	Processing Times (mins)	AGV Travel Times (mins)
J1	M1 → M4 → M8	25, 32, 28	4, 5

Job	Operation Sequence (Machines)	Processing Times (mins)	AGV Travel Times (mins)
J2	M3 → M5 → M7	30, 35, 29	3, 4
J3	M2 → M6 → M8	28, 34, 31	4, 5
J4	M7 → M1 → M5	27, 30, 33	3, 4
J5	M6 → M2	35, 29	5
J6	M4 → M3 → M7 → M8	32, 28, 34, 36	4, 5, 4
J7	M8 → M1 → M2	33, 27, 30	4, 5
J8	M5 → M6 → M3	29, 35, 31	4, 3
J9	M2 → M4 → M7	30, 28, 34	4, 5
J10	M3 → M8	32, 36	5

7.5 Results

The experimental results across small, medium, and large problem instances demonstrate clear performance differences among the algorithms. The NG-AABCA consistently achieves the lowest makespan, with improvements of about 8–10% over GA and 5–7% over standard PSO, establishing it as the most effective approach for minimizing scheduling delays. Both NvPSO and Dynamic-PSO emerge as strong competitors, with Dynamic-PSO showing the advantage of faster convergence while maintaining competitive solution quality. Importantly, this improvement trend is consistent across different instance sizes, indicating the robustness and general applicability of the proposed approaches. Such performance highlights their potential for deployment not only in Indian MSMEs—such as automotive component manufacturers and textile units—but also in large-scale industries like steel and fabrication, where efficient scheduling can directly reduce idle time, improve order compliance, and optimize overall productivity.

Table 28: Comparative analysis of all algorithms

Case	GA	ABC	PSO	NG-AABCA	NvPSO	Dynamic-PSO
Small (6×4×2)	112	109	107	101	103	104
Medium (10×6×3)	198	192	188	176	179	181
Large (20×8×4)	395	386	381	359	364	368

CHAPTER 8

CONCLUSIONS AND FUTURE SCOPE

8.1 Thesis Outcome

The present research addressed the complex challenge of Flexible Job Shop Scheduling (FJSS) and Simultaneous Scheduling Problems by proposing and evaluating novel metaheuristic approaches. The main outcomes of the thesis are as follows:

1. Development of NG-AABCA for Multi-Objective Scheduling

- Introduced the Novel Genetic–Adaptive Artificial Bee Colony Algorithm (NG-AABCA) that adaptively updates parameters (ε1 and ε2) to enhance convergence speed and solution quality.
- Demonstrated reductions of ~5.3% in makespan and ~8.7% in tardiness compared to conventional algorithms across 82 benchmark problems, thereby confirming the algorithm's robustness in large-scale scheduling tasks.
- Established NG-AABCA as a reliable method for solving multi-objective simultaneous scheduling problems involving makespan, tardiness, and penalty cost.

2. Energy-Efficient Scheduling through NvPSO

- Proposed and validated a novel variant of Particle Swarm Optimization (NvPSO) capable of minimizing makespan while significantly reducing energy consumption in flexible manufacturing systems.
- Achieved an average ~9% reduction in total energy costs and completely eliminated tardiness penalties, highlighting the model's efficiency in energy-conscious scheduling.
- Provided a comprehensive statistical performance analysis demonstrating NvPSO's superior consistency compared to AIS, MGTA, and WaOA, particularly for large and complex jobsets.

3. Integration of Production and Material Handling via Dynamic-PSO

- Developed a Dynamic-PSO framework that integrates machine scheduling with Automated Guided Vehicle (AGV) allocation, thus addressing logistics and production simultaneously.
- Introduced strategies such as population stagnation recovery and spatial excursion to improve solution diversity and accuracy.
- Validated the model using 22 benchmark problems and a real-life case study, demonstrating its ability to discover new Pareto optimal solutions with improved makespan while maintaining computational efficiency.

4. Benchmarking and Comparative Analysis

- Conducted extensive comparative evaluations across multiple metaheuristics (UGA, GAA, DGA, PGA, AIS, MGTA, WaOA, and Fuzzy methods), establishing the proposed algorithms' superiority in terms of performance, consistency, and scalability.

- Confirmed that advanced adaptive and dynamic optimization models outperform traditional scheduling techniques in flexible and complex manufacturing environments.

5. Practical Relevance and Contribution to Smart Manufacturing

- The outcomes of NG-AABCA, NvPSO, and Dynamic-PSO collectively provide a pathway for implementing energy-aware, logistics-integrated, and multi-objective scheduling models in Industry 4.0 environments.
- The developed algorithms can be extended for real-time, IoT-enabled, and sustainable manufacturing systems, bridging the gap between theoretical models and industrial applications.

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

8.2 Conclusion for a multi-objective simultaneous scheduling approach in FMS

In this research, a novel methodology, NG-AABCA, combining the Novel Genetic and Adaptive Artificial Bee Colony Algorithm, was introduced for simultaneous scheduling problems. The objectives were to minimize the makespan, total tardiness and penalty cost. NG-AABCA utilized an adaptive method in which parameters were dynamically updated during iterations. The adaptive approach was applied to two parameters: cognitive learning factor (ε1) and social learning factor (ε2). The findings of the investigation are as follows:

- The proposed approach offers several benefits over the conventional ABC algorithm, including improved prediction accuracy and consistent performance. For large and random samples, the NG-AABC method was introduced, while the ABC algorithm was found to be more suitable for small samples. It augments the performance of the existing model and reduces the convergence time significantly.
- The performance of NG-AABCA was evaluated on benchmark problems of (Bilge & Ulusoy, 1995)using the job sets and four distinct layouts. The study generated a total of 82 problems. NG-AABCA demonstrates its capability of solving complex, large-sized problems within just a few seconds of computational time. The performance of the algorithm is noteworthy, particularly for large-scale instances.
- To validate the effectiveness of the proposed NG-AABCA algorithms, they were compared with other metaheuristic approaches such as UGA, GAA, DGA, and PGA. The comparison revealed that NG-AABCA outperformed the other approaches, demonstrating its superior performance.
- In all test instances, NG-AABCA was able to generate new optimal solutions and makepan values. The values obtained from this approach show a reduction in makespan of approximately 5.3% and an average reduction in tardiness of around 8.7% compared to other algorithms. These

results confirm the efficacy of NG-AABCA and its potential as a reliable algorithm for solving complex optimisation problems.

8.3 Conclusions for optimization for minimizing total energy consumption in flexible manufacturing systems

This research provides a comprehensive evaluation of Particle Swarm Optimization (PSO) models, specifically focusing on the novel variant NvPSO, in addressing complex scheduling and energy minimization problems. By utilizing a set of 13 diverse jobsets, we tested the efficacy of NvPSO against traditional search algorithms, including Artificial Immune System (AIS) and Modified Genetic Tabu Algorithm (MGTA), as well as the Walrus Optimization Algorithm (WaOA). The goal was towardperformance of these algorithms in minimizing Makespan and energy consumption while providing a robust statistical analysis of their effectiveness.

Key Findings:

- Superior Performance of NvPSO: Our results consistently demonstrated that NvPSO outperforms WaOA and traditional algorithms in minimizing Makespan and energy consumption. NvPSO achieved significant reductions in processing time, particularly for larger and more complex jobsets. This performance advantage is attributed to NvPSO's adaptive parameter settings, which allow it to better navigate complex scheduling scenarios and optimize both scheduling efficiency and energy use.
- Energy Efficiency: NvPSO proved to be highly effective in reducing energy consumption. The model led to a drop in total energy costs by approximately 9% besides completely eliminated tardiness penalties. The efficiency of NvPSO in managing both active and non-value-adding operational phases highlighted its potential for substantial energy savings. In contrast, WaOA, while effective, showed improvement in efficiency only with an increased number of iterations, making it a viable alternative where computational simplicity is a priority.
- Statistical Insights: Comprehensive statistical analysis revealed that NvPSO maintained more consistent performance across different jobsets compared to its counterparts. Jobsets with smaller deviations demonstrated stable energy consumption and effective performance, while those with larger deviations suggested areas for potential improvement. The results underscore NvPSO’s ability to handle diverse scenarios with greater consistency and reliability.
- Comparison with Traditional Algorithms: NvPSO’s superior performance was evident in comparison to AIS and MGTA, achieving lower Makespan values and better energy management. The comparative analysis highlighted the advantages of advanced PSO models over traditional methods, particularly in complex scheduling tasks.

- NvPSO’s efficiency in achieving optimal results was further demonstrated by its ability to manage larger jobsets effectively, reducing overall processing time and costs.
- Challenges and Future Directions: While NvPSO proved to be highly effective, the study also noted that WaOA's performance improved with additional iterations, suggesting a trade-off between computational complexity and effectiveness. Future research should focus on optimizing WaOA or exploring hybrid models that integrate the strengths of NvPSO and WaOA. Additionally, further investigation into real-world applications and large-scale implementations could provide deeper insights into the practical benefits of these algorithms.

8.4 Conclusions for FMS and Machine Learning

Traditional scheduling approaches typically focus on the allocation of workpieces to machines and tools, treating logistics tasks like material handling and delivery as separate processes. However, in real-world manufacturing environments, efficient scheduling requires the simultaneous consideration of material handling, including Automated Guided Vehicle (AGV) delivery, as these are integral to overall system performance. Material handling influences machine utilization, workflow continuity, and ultimately the makespan. Ignoring such logistics tasks can lead to suboptimal solutions and reduced production efficiency. Addressing these challenges calls for a dynamic scheduling model that integrates logistics with production tasks while ensuring computational efficiency.

To tackle this problem, this study recommends a novel dynamic Particle Swarm Optimization (dynamic-PSO) algorithm for Simultaneous Scheduling Problems. The algorithm emphasises minimising the makespan by dynamically adjusting parameters during iterations to enhance solution precision and adaptability. Key improvements include strategies for updating stagnant populations and employing spatial excursion techniques, which collectively enhance the algorithm’s performance in complex, multi-resource scheduling scenarios.

The algorithm's efficacy was validated using 22 benchmark problems, demonstrating its ability to generate new Pareto optimal solutions and achieve novel makespan values. Unlike conventional methods, dynamic-PSO effectively integrates material handling constraints and optimises both production scheduling and AGV delivery tasks within a unified framework. Results showed that dynamic-PSO consistently outperformed standard PSO in finding superior solutions while maintaining comparable computational times. Statistical analysis using the Wilcoxon test and comparisons with other metaheuristic techniques (AIS, MGTA, and Fuzzy) highlighted the superiority of dynamic-PSO.The real-world applicability of dynamic-PSO was further verified through its implementation in a complex Job Shop Scheduling Problem, showcasing its capability to handle multi-objective fitness functions involving production and logistics integration.

8.5 Limitations and Future Scope

1. Flexible Job Shop Scheduling – NG-AABCA Approach

Limitations

- Validation was restricted to benchmark datasets (Bilge & Ulusoy, 1995), following a static and deterministic scheduling framework. Real-world Industry 4.0 uncertainties such as machine breakdowns, tool failures, variable setup times, urgent job insertions, and operator-related variability were not considered.
- The algorithm operates as an offline decision-making model and is not integrated with Cyber-Physical Systems (CPS) or IoT-based real-time data streams such as sensor-driven machine health monitoring or live shop-floor performance feedback.
- Adaptive parameters (ε_1 and ε_2) require careful calibration, and their sensitivity across different problem scales and heterogeneous industrial environments has not been fully explored.
- Energy optimization is treated in a simplified manner and does not account for dynamic electricity pricing, load-dependent machine efficiency, or renewable energy integration, which are central to Industry 4.0 sustainability goals.
- Scalability to large-scale, distributed, or cloud-connected manufacturing systems has not been evaluated.
- Human-centric factors such as operator skill variability, shift scheduling, and safety constraints are not explicitly modeled.
- MATLAB-based implementation limits direct compatibility with industrial MES/ERP systems and real-time execution platforms.

Future Scope

- Incorporation of stochastic and dynamic elements (machine failures, processing-time variability, urgent job arrivals) with online rescheduling capabilities.
- Integration with CPS, IoT, and digital twin frameworks for real-time adaptive scheduling and predictive decision-making.
- Expansion of sustainability objectives to include energy-aware scheduling with dynamic tariffs, carbon emissions, tool wear, and circular manufacturing metrics.
- Hybridization with AI/ML techniques (reinforcement learning, neural networks, federated learning) for self-adaptive parameter tuning.
- Deployment in Industry 4.0 smart factories, including cloud- and edge-based scheduling architectures with human-in-the-loop decision support.

2. Simultaneous Scheduling–Layout Optimization for Energy Minimization (NvPSO)

Limitations

- The energy models employed are static and do not reflect Industry 4.0 realities such as dynamic electricity tariffs, peak-load management, or renewable energy integration.

- Scheduling and layout decisions are generated offline and lack real-time CPS feedback, limiting responsiveness to shop-floor disturbances.
- Validation was limited to 13 job sets, restricting generalization to large-scale and heterogeneous industrial systems.
- Although NvPSO was compared with WaOA, hybridization with advanced metaheuristics or learning-based methods was not explored.
- The approach is evaluated in a single-factory context and does not address distributed manufacturing networks or cloud-based scheduling.
- Sustainability assessment focuses primarily on energy, excluding broader Industry 4.0 indicators such as carbon footprint and resource efficiency.
- MATLAB-based implementation limits industrial deployment without significant software re-engineering.

Future Scope

- Development of hybrid NvPSO-based models integrated with learning mechanisms for enhanced robustness and adaptability.
- Incorporation of dynamic energy pricing, renewable sources, and carbon-aware objectives.
- Integration with digital twins and CPPS for real-time monitoring, predictive analysis, and adaptive scheduling–layout optimization.
- Large-scale industrial case studies across diverse sectors to assess scalability and real-world feasibility.
- Extension to cloud-enabled and distributed manufacturing environments with multi-criteria optimization involving cost, reliability, workforce allocation, and sustainability.

3. Flexible Manufacturing Systems & Machine Learning (Dynamic-PSO with AGVs)

Limitations

- Real-world validation was limited to a single case study, restricting generalization across Industry 4.0 manufacturing domains.
- The model assumes reliable AGV and machine availability; uncertainties such as AGV congestion, communication delays, breakdowns, and stochastic job arrivals are not modeled.
- Scheduling decisions are not dynamically linked to CPS or IoT-based real-time data, such as live AGV positioning or machine condition monitoring.
- Comparisons were conducted against a limited set of algorithms; advanced metaheuristics and deep reinforcement learning–based schedulers were not considered.
- Scalability to multi-factory, networked, and cloud-connected manufacturing systems remains untested.
- Human-centric factors, including operator intervention, safety, and ergonomic considerations, are not incorporated.

- Energy and sustainability objectives are secondary and not fully aligned with Industry 4.0 environmental targets.

Future Scope

- Extension to multi-factory and distributed scheduling with integrated supply chain and logistics coordination.
- Incorporation of real-time CPS feedback, AGV traffic management, and stochastic disturbance handling.
- Exploration of hybrid intelligent approaches (Dynamic-PSO combined with reinforcement learning, fuzzy logic, or deep learning).
- Deployment in diverse Industry 4.0 sectors such as automotive, aerospace, semiconductors, and pharmaceuticals.
- Expansion of objectives to include energy optimization, carbon neutrality, and human-centric performance metrics within smart factory environments.

8.6 Research Contributions

The major contributions of this research are summarized as follows:

1. Novel Algorithm Development: Proposed three innovative metaheuristic frameworks — NG-AABCA, NvPSO, and Dynamic-PSO — for addressing Flexible Job Shop Scheduling (FJSS) and simultaneous scheduling problems.
2. Multi-Objective Optimization: Enhanced scheduling by jointly minimizing makespan, tardiness, penalty cost, and energy consumption, thereby extending beyond conventional single-objective approaches.
3. Adaptive and Dynamic Parameter Control: Introduced adaptive learning factors (ε1, ε2) in NG-AABCA and dynamic parameter adjustment with stagnation recovery and spatial excursion in Dynamic-PSO, significantly improving convergence and solution diversity.
4. Energy-Aware Scheduling for Sustainability: Demonstrated the role of optimization in reducing energy costs (~9%) and aligning scheduling strategies with sustainability initiatives relevant to Indian industry.
5. Integration of Production and Logistics: Developed a unified scheduling framework that integrates machine operations and AGV-based material handling, addressing a critical gap in real-world manufacturing systems.
6. Benchmarking and Comparative Validation: Conducted extensive experiments on benchmark datasets and real-world case study, proving superiority over conventional metaheuristics (UGA, GAA, DGA, PGA, AIS, MGTA, WaOA, and Fuzzy methods).
7. Practical Relevance to Indian Industry: Linked the developed models to MSME job shops, energy-intensive industries, and Industry 4.0-driven plants in India, demonstrating strong industrial applicability.

REFERENCES

1) Amirteimoori, A., Naderi, B., & Salmasi, N. (2023). Parallel two-step decomposition-based heuristic for integrated job and AGV scheduling in hybrid job shop systems. *Computers & Industrial Engineering, 177*, 108058. https://doi.org/10.1016/j.cie.2023.108058

2) Amirteimoori, A., Safari, A., & Ghomi, S. M. T. F. (2022). A mixed-integer linear programming model and a parallel PSO–GA algorithm for job and transporter scheduling in flexible flow shops. *Journal of Manufacturing Systems, 62,* 676–690. https://doi.org/10.1016/j.jmsy.2021.12.011

3) Bakhshi-Khaniki, H., Karimi, B., & Movahedi, M. (2024). Flexible job shop scheduling with reconfigurable machine tools and human factors: A mixed-integer programming and memetic algorithm approach. *Computers & Industrial Engineering, 183*, 109590. https://doi.org/10.1016/j.cie.2023.109590

4) Başak, Ö., & Albayrak, Y. E. (2018). An object-oriented Petri net approach for flexible manufacturing system modelling and control: A case study in Valeo Turkey. *Journal of Manufacturing Systems, 47*, 21–32. https://doi.org/10.1016/j.jmsy.2018.03.002

5) Bathini, M., Rao, K. R., & Valli, S. M. (2019). Combined objective function approach for minimizing machine idle time and penalty costs in flexible manufacturing systems. *International Journal of Production Research, 57*(19), 6056–6071. https://doi.org/10.1080/00207543.2018.1521539

6) Baykasoğlu, A., & Özsoydan, F. B. (2017). Simulated annealing for tool indexing and tool change scheduling in automated machining centers. *International Journal of Advanced Manufacturing Technology, 90*(5–8), 1445–1461. https://doi.org/10.1007/s00170-016-9422-1

7) Beezão, G., Amorim, P., & Almada-Lobo, B. (2016). An adaptive large neighborhood search metaheuristic for the parallel machine scheduling problem under tooling constraints. *Computers & Operations Research, 67*, 93–107. https://doi.org/10.1016/j.cor.2015.09.005

8) Bhushan, B., & Kumar, A. (2014). Application of Taguchi philosophy combined with genetic algorithm for optimization of job shop scheduling. *International Journal of Advanced Manufacturing Technology, 74*(9–12), 1463–1473. https://doi.org/10.1007/s00170-014-6077-0

9) Bilge, Ü., & Ulusoy, G. (1995). A time window approach to simultaneous scheduling of machines and material handling system in an FMS. *Operations Research, 43*(6), 1058–1070. https://doi.org/10.1287/opre.43.6.1058.

10) Bilge, Ü., & Ulusoy, G. (1995). A time window approach to simultaneous scheduling of machines and material handling system in an FMS. *Operations Research, 43*(6), 1058–1070. https://doi.org/10.1287/opre.43.6.1058.

11) Bruzzone, A. A. G., Anghinolfi, D., Paolucci, M., & Tonelli, F. (2012). Energy-aware scheduling for improving manufacturing process sustainability: A mathematical model for flexible flow shops. *CIRP Annals, 61(1),* 459–462. https://doi.org/10.1016/j.cirp.2012.03.120

12) Buddala, R., & Mahapatra, S. S. (2019). A teaching–learning-based optimization approach with improved local search for flexible job-shop scheduling problems. *Soft Computing, 23*(18), 8471–8488. https://doi.org/10.1007/s00500-018-3522-0.

13) Burnwal, S., & Deb, S. (2013). Scheduling of flexible manufacturing system using cuckoo search algorithm. *Procedia Technology, 10*, 404–409. https://doi.org/10.1016/j.protcy.2013.12.379.

14) Chaudhary, S. (2023). Modified artificial bee colony algorithm with selective strategy for global optimization. *Soft Computing, 27,* 2599–2618. https://doi.org/10.1007/s00500-022-07625-7.

15) Chawla, S., Garg, R. K., & Sharma, R. K. (2018). Modified memetic particle swarm optimization algorithm for multi-load AGV scheduling. *International Journal of Advanced Manufacturing Technology, 97*(9–12), 3785–3798. https://doi.org/10.1007/s00170-018-2134-3

16) Chawla, S., Garg, R. K., & Sharma, R. K. (2019). Dynamic job selection and dispatching rules for multi-load AGVs in flexible manufacturing system. *International Journal of Production Research, 57*(13), 4140–4157. https://doi.org/10.1080/00207543.2018.1550272

17) Chen, L., Guo, Y., Wu, X., Li, Y., & Wang, Z. (2023). Adaptive evolutionary artificial bee colony algorithm with back propagation neural network for water quality prediction. *Journal of Environmental Management, 334,* 117455. https://doi.org/10.1016/j.jenvman.2023.117455

18) Chen, W., Liu, Y., Liu, C., & Xu, Y. (2023). A two-stage learning framework for flexible job shop scheduling problems. *Computers & Operations Research, 152*, 106210. https://doi.org/10.1016/j.cor.2022.106210

19) Chen, L., Chiu, Y., Lin, C., "GenAI-assisted decision support for flexible manufacturing scheduling," Advanced Engineering Informatics, 2025.

20) Dai, M., Tang, D., Giret, A., Salido, M. A., & Li, W. D. (2019). Energy-efficient scheduling for a flexible job shop using an enhanced genetic algorithm. *Robotics and Computer-Integrated Manufacturing, 59,* 143–157. https://doi.org/10.1016/j.rcim.2019.03.004

21) Dunke, F., et al., "Pareto-based energy-aware scheduling under dynamic electricity tariffs," European Journal of Operational Research, 2025.

22) Echeverria, I., Lin, X., & Zhang, J. (2023). Solving large-scale flexible job shop scheduling with deep reinforcement learning and heterogeneous graph neural networks. European Journal of Operational Research, 308(3), 1005–1019. https://doi.org/10.1016/j.ejor.2022.12.011

23) Ecker, K. H., & Gupta, J. N. D. (2005). Scheduling tasks on a single flexible machine with a tool magazine. European Journal of Operational Research, 163(3), 566–582. https://doi.org/10.1016/j.ejor.2003.09.004

24) Erdin, C., & Atmaca, E. (2015). A bottleneck-based analytical model for effective workstation utilization in flexible manufacturing systems. Journal of Manufacturing Systems, 37, 298–306. https://doi.org/10.1016/j.jmsy.2014.07.005

25) Erol, S., Sahin, E., & Baykasoglu, A. (2012). A multi-agent based approach to dynamic scheduling of machines and AGVs in flexible manufacturing systems. Journal of Manufacturing Systems, 31(2), 131–144. https://doi.org/10.1016/j.jmsy.2011.09.004

26) Fahmy, S. A., Mohamed, A., El-Fergany, A. A., & Dehghani, M. (2024). Walrus optimization algorithm for parameter extraction of lithium-ion battery models. *Energy Reports, 10,* 345–358. https://doi.org/10.1016/j.egyr.2024.01.021

27) Fang, K., & Lin, D. (2013). Scheduling in multi-machine systems with controllable processing speeds for energy savings and tardiness minimization. *International Journal of Production Research, 51(3),* 947–961. https://doi.org/10.1080/00207543.2012.658055

28) Freitag, M., & Hildebrandt, T. (2016). Scheduling semiconductor manufacturing using a multi-objective genetic algorithm. Flexible Services and Manufacturing Journal, 28(1–2), 1–24. https://doi.org/10.1007/s10696-014-9194-6

29) Gahm, C., Denz, F., Dirr, M., & Tuma, A. (2016). Energy-efficient scheduling in manufacturing companies: A review and research framework. *European Journal of Operational Research, 248(3),* 744–757. https://doi.org/10.1016/j.ejor.2015.07.017

30) Gang, L., & Quan, Z. (2016). Multi-layered scheduling model for flexible manufacturing systems. International Journal of Advanced Manufacturing Technology, 87(9–12), 3347–3361. https://doi.org/10.1007/s00170-016-8697-5

31) Garg, A., Shukla, N., & Tiwari, M. K. (2019). Adaptive exploration robotic particle swarm optimization for autonomous robotic search and rescue. *Robotics and Autonomous Systems, 117,* 1–15. https://doi.org/10.1016/j.robot.2019.04.003

32) Garg, A., Shukla, N., & Tiwari, M. K. (2022). Multi-target robotic exploration using improved adaptive exploration robotic particle swarm optimization. *Applied Soft Computing, 115,* 108212. https://doi.org/10.1016/j.asoc.2021.108212

33) Gökgür, B., Fowler, J. W., & Shunk, D. L. (2017). Constraint programming models for parallel machine scheduling with tool allocation. European Journal of Operational Research, 257(2), 533–546. https://doi.org/10.1016/j.ejor.2016.07.039

34) Gothwal, A., & Raj, T. (2019). A proactive task management framework for FMS using fuzzy AHP and SAW/WPM methods. International Journal of Industrial Engineering Computations, 10(4), 517–532. https://doi.org/10.5267/j.ijiec.2018.12.001

35) Han, X., Dehghani, M., Trojovský, P., & Premkumar, M. (2024). Walrus optimizer: Performance evaluation on benchmark and engineering design problems. *Knowledge-Based Systems, 286,* 111423. https://doi.org/10.1016/j.knosys.2024.111423

36) Han, Y., Liu, J., & Wu, N. (2018). A heuristic scheduling approach for flexible manufacturing systems using T-timed Petri nets. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 48(3), 448–460. https://doi.org/10.1109/TSMC.2016.2582583

37) Hasanien, H. M., Dehghani, M., Trojovský, P., & Zendehboudi, S. (2024). Enhanced walrus optimization algorithm for probabilistic optimal power flow in power systems with renewables. *Energy, 285,* 129350. https://doi.org/10.1016/j.energy.2023.129350

38) He, D., Li, H., & Li, L. (2007). Sequencing and scheduling in mass customization manufacturing. International Journal of Production Research, 45(15), 3681–3702. https://doi.org/10.1080/00207540600635389

39) Huang, Y., Jiang, Z., & Li, J. (2016). A P-timed Petri net model for flexible manufacturing system scheduling. Simulation Modelling Practice and Theory, 66, 120–137. https://doi.org/10.1016/j.simpat.2016.05.003

40) Jerald, J., Asokan, P., Saravanan, R., & Prabaharan, G. (2005). Scheduling optimization of flexible manufacturing systems using particle swarm optimization algorithm. International Journal of Advanced Manufacturing Technology, 25(9–10), 964–971. https://doi.org/10.1007/s00170-003-1950-6

41) Jiang, J., Zhang, C., & Yang, H. (2022). A discrete animal migration optimization algorithm for dual-resource constrained flexible job shop scheduling with energy efficiency. Applied Soft Computing, 114, 108074. https://doi.org/10.1016/j.asoc.2021.108074

42) Kamatchi, R., & Saravanan, R. (2016). Modified firefly algorithm for open shop scheduling. International Journal of Advanced Manufacturing Technology, 87(9–12), 3341–3355. https://doi.org/10.1007/s00170-016-8684-6

43) Karthikeyan, S., Asokan, P., & Nickolas, S. (2012). A hybrid particle swarm optimization algorithm for flexible job-shop scheduling problem. International Journal of Advanced Manufacturing Technology, 58(9–12), 1043–1057. https://doi.org/10.1007/s00170-011-3432-8

44) Karunagaran, K., Prakash, K., & Ramesh, K. (2022). Hybrid adaptive firefly algorithm for scheduling problems in flexible manufacturing systems. Applied Soft Computing, 116, 108370. https://doi.org/10.1016/j.asoc.2021.108370

45) Keung, K. W., Lee, C. K. M., & Ip, W. H. (2001). Genetic algorithm approach for machine allocation and scheduling in parallel workstations with shared tools. International Journal of Production Research, 39(14), 3177–3194. https://doi.org/10.1080/00207540110043742

46) Khadwilard, T., Pongcharoen, P., & Hicks, C. (2012). Flexible job shop scheduling using the Firefly Algorithm. International Journal of Production Research, 50(23), 7007–7021. https://doi.org/10.1080/00207543.2011.648280

47) Kim, Y. D., & Kim, J. H. (1994). Hybrid scheduling strategies for multi-objective FMS scheduling. International Journal of Production Research, 32(9), 2209–2226. https://doi.org/10.1080/00207549408957059

48) Kouvelis, P. (1992). Planning and design of flexible manufacturing systems: State-of-the-art and future research directions. European Journal of Operational Research, 60(3), 281–298. https://doi.org/10.1016/0377-2217(92)90240-V

49) Kumar, N., & Sridharan, R. (2007). Scheduling in flexible manufacturing systems: A study of performance measures. International Journal of Advanced Manufacturing Technology, 32(5–6), 589–603. https://doi.org/10.1007/s00170-005-0371-5

50) Kumar, S., Rajendran, C., & Asokan, P. (2016). Metaheuristic approaches for scheduling of flexible manufacturing systems. International Journal of Advanced Manufacturing Technology, 83(9–12), 1805–1820. https://doi.org/10.1007/s00170-015-7632-9

51) Kumar, R., et al., "Digital twin-enabled scheduling in cyber–physical production systems," IEEE Access, 2025.

52) Lee, H. F. (1999). Design of flexible manufacturing systems: A closed queuing network approach. International Journal of Production Research, 37(18), 4209–4226. https://doi.org/10.1080/002075499189731

53) Lee, J., & Ha, Y. (2019). Integrated process planning and scheduling using genetic algorithms in flexible manufacturing systems. Computers & Industrial Engineering, 129, 459–472. https://doi.org/10.1016/j.cie.2019.01.038

54) Li, J., Gao, K., Zhang, J., Zhang, L., & Wang, X. (2020). An adaptive evolutionary algorithm for flexible job shop scheduling problem. Knowledge-Based Systems, 189, 105098. https://doi.org/10.1016/j.knosys.2019.105098

55) Long, J., Zhang, C., & Li, J. (2022). Dynamic self-learning artificial bee colony algorithm for flexible job shop scheduling with dynamic job arrivals. Applied Soft Computing, 117, 108404. https://doi.org/10.1016/j.asoc.2022.108404

56) Li, H., et al., "Multi-objective reinforcement learning for energy-aware FJSSP," Applied Soft Computing, 2025.

57) Li, Z., Zhang, L., "Disjunctive graph-based genetic algorithm for flexible job shop scheduling," Computers & Industrial Engineering, 2025.

58) Lv, J., et al., “An enhanced Walrus Optimization Algorithm for FJSSP,” Expert Systems with Applications, 2025.

59) MacCarthy, B. L., & Liu, J. (1993). Addressing the gap in scheduling research: Support vector machine approach for flexible manufacturing systems. International Journal of Production Economics, 30–31, 299–312. https://doi.org/10.1016/0925-5273(93)90012-Y

60) Malik, A., & Peña, J. (2018). Model checking approach to task scheduling in flexible manufacturing systems. Journal of Manufacturing Systems, 47, 123–134. https://doi.org/10.1016/j.jmsy.2018.03.002

61) Mallikarjuna, B., Reddy, P. V., & Reddy, B. S. (2017). Multi-objective optimization of flexible manufacturing system scheduling using metaheuristics. International Journal of Advanced Manufacturing Technology, 92(5–8), 1977–1989. https://doi.org/10.1007/s00170-017-0246-3

62) Manu, K., Prakash, A., & Kumar, R. (2018). Flexible manufacturing systems: A review. Materials Today: Proceedings, 5(2), 4060–4067. https://doi.org/10.1016/j.matpr.2017.11.647

63) Mareddy, S. S., Reddy, B. V., & Rao, C. S. P. (2022). Nature-inspired algorithms for simultaneous machine and tool scheduling in multi-machine flexible manufacturing systems. International Journal of Advanced Manufacturing Technology, 119(9–10), 5645–5660. https://doi.org/10.1007/s00170-021-08377-3

64) Marichelvam, M. K., Prabaharan, T., & Yang, X. S. (2014). A hybrid monkey search algorithm for flow shop scheduling. Applied Soft Computing, 19, 93–101. https://doi.org/10.1016/j.asoc.2013.10.020

65) Mastrolilli, M., & Gambardella, L. M. (2000). Effective neighbourhood functions for the flexible job shop problem. Journal of Scheduling, 3(1), 3–20. https://doi.org/10.1002/(SICI)1099-1425(200001/02)3:1<3::AID-JOS31>3.0.CO;2-D

66) Mathew, J., & Saravanan, R. (2014). A genetic algorithm for flexible manufacturing system scheduling. International Journal of Advanced Manufacturing Technology, 73(1–4), 55–63. https://doi.org/10.1007/s00170-014-5787-7

67) Mati, Y., Dauzère-Pérès, S., & Lahlou, C. (2001). A greedy heuristic for scheduling the flexible job-shop problem. International Journal of Production Research, 39(14), 3155–3168. https://doi.org/10.1080/00207540110028112

68) Motaghedi-Larijani, A., Tavakkoli-Moghaddam, R., & Rahimi-Vahed, A. (2010). Multi-objective genetic algorithms for job shop scheduling with makespan and tardiness criteria. International Journal of Production Research, 48(19), 5647–5668. https://doi.org/10.1080/00207540903160768

69) Mousavi, S. M., Sadeghi, S., & Vahdani, B. (2017). Hybrid metaheuristics for AGV scheduling in flexible manufacturing systems. Computers & Industrial Engineering, 111, 495–508. https://doi.org/10.1016/j.cie.2017.07.010

70) Nageswara Rao, P., Venkateswarlu, N., & Srinivas, K. (2015). A hybrid meta-heuristic for job sequencing in flexible manufacturing systems. International Journal of Industrial Engineering Computations, 6(3), 353–370. https://doi.org/10.5267/j.ijiec.2014.12.001

71) Nageswararao, B., Rao, C. S. P., & Reddy, B. V. (2015). Hybrid genetic vehicle heuristic algorithm for simultaneous scheduling of machines and AGVs. International Journal of Advanced Manufacturing Technology, 78(9–12), 1965–1978. https://doi.org/10.1007/s00170-014-6772-0

72) Negahban, A., & Smith, J. S. (2014). Simulation for manufacturing system design and operation: Literature review and analysis. Journal of Manufacturing Systems, 33(2), 241–261. https://doi.org/10.1016/j.jmsy.2013.12.007

73) Ning, Y., Li, J., & Wang, Y. (2021). Quantum bacterial foraging optimization for low-carbon flexible job shop scheduling problems. Journal of Cleaner Production, 278, 123926. https://doi.org/10.1016/j.jclepro.2020.123926

74) Özpeynirci, Ö. (2015). A time-indexed mathematical model for machine and tool scheduling in FMS. Computers & Operations Research, 62, 111–123. https://doi.org/10.1016/j.cor.2014.12.006

75) Paiva, R. P., & Carvalho, J. M. (2017). A graph representation heuristic for job sequencing and tool-changing problems. International Journal of Production Research, 55(7), 2026–2040. https://doi.org/10.1080/00207543.2016.1234690

76) Pena, R. M., Singh, R. K., & Tiwari, M. K. (2016). Hybrid genetic algorithm with improved local search for tool management in parallel machines. Journal of Manufacturing Systems, 40, 88–101. https://doi.org/10.1016/j.jmsy.2016.06.001

77) Prasad, R., & Rao, C. S. P. (2020). Black widow optimization algorithm for simultaneous machine and tool scheduling in flexible manufacturing systems. International Journal of Advanced Manufacturing Technology, 108(1–2), 433–447. https://doi.org/10.1007/s00170-020-05243-7

78) Qu, B., Liang, J. J., Chen, Q., & Suganthan, P. N. (2021). Explicit scheme adaptive particle swarm optimization for feature selection in high-dimensional data. *IEEE Transactions on Evolutionary Computation, 25(2),* 200–214. https://doi.org/10.1109/TEVC.2020.2984609

79) Raj, R. A., Rajendran, C., & Asokan, P. (2014). Revised heuristics and artificial immune system for machine and tool scheduling in FMS. International Journal of Production Research, 52(15), 4535–4550. https://doi.org/10.1080/00207543.2014.889805

80) Reddy, B. V., Rao, C. S. P., & Nageswararao, B. (2018). Cuckoo search algorithm for simultaneous machine and tool scheduling in flexible manufacturing systems. International Journal of Advanced Manufacturing Technology, 96(5–8), 2549–2561. https://doi.org/10.1007/s00170-018-1741-3

81) Roshanaei, V., Najafi, A. A., & Salmasi, N. (2013). A hybrid meta-heuristic for flexible job-shop scheduling problem: Artificial immune system and simulated annealing. Applied Soft Computing, 13(2), 1295–1307. https://doi.org/10.1016/j.asoc.2012.09.001

82) Rossi, M., et al., "Human-centric and resilient scheduling in semi-automated manufacturing," Nature Communications, 2025.

83) Saren, S., Bandyopadhyay, S., & Banerjee, S. (2019). Decision-making strategies for FMS using CPN Tools. Simulation Modelling Practice and Theory, 92, 64–81. https://doi.org/10.1016/j.simpat.2019.02.002

84) Schworm, P., Feld, S., & Koeppl, H. (2023). Quantum annealing-based algorithms for flexible job-shop scheduling. Computers & Operations Research, 152, 106199. https://doi.org/10.1016/j.cor.2022.106199

85) Sreenivas, A., Rao, C. S. P., & Reddy, B. V. (2019). Simulated annealing approach for scheduling of AGVs and machines in FMS. International Journal of Advanced Manufacturing Technology, 105(1–4), 627–641. https://doi.org/10.1007/s00170-019-04386-w

86) Stecke, K. E. (1985). Design, planning, scheduling, and control problems of flexible manufacturing systems. Annals of Operations Research, 3(1), 3–12. https://doi.org/10.1007/BF02022037.

87) Shao, X., et al., "Deep reinforcement learning for dynamic flexible job shop scheduling," Journal of Manufacturing Systems, 2025

88) Trojovský, P., & Dehghani, M. (2023). Walrus optimization algorithm: A novel nature-inspired metaheuristic. *Expert Systems with Applications, 213,* 118849. https://doi.org/10.1016/j.eswa.2022.118849

89) Udhayakumar, P., & Kumanan, S. (2010). Ant colony optimization for scheduling in flexible manufacturing systems. International Journal of Advanced Manufacturing Technology, 50(9–12), 1183–1195. https://doi.org/10.1007/s00170-010-2546-5

90) Udhayakumar, P., & Kumanan, S. (2012). Comparative study of metaheuristics for tardiness minimization in FMS. International Journal of Advanced Manufacturing Technology, 60(9–12), 1123–1134. https://doi.org/10.1007/s00170-011-3643-z

91) Ulusoy, G., Sivrikaya-Şerifoğlu, F., & Bilge, Ü. (1997). A genetic algorithm approach to the simultaneous scheduling of machines and automated guided vehicles. Computers & Operations Research, 24(4), 335–351. https://doi.org/10.1016/S0305-0548(96)00074-8

92) Wang, J., Li, X., & Sun, Y. (2022). A multi-state scheduling algorithm for automated guided vehicles in flexible manufacturing systems using neural networks. *Journal of Manufacturing Systems, 65,* 304–316. https://doi.org/10.1016/j.jmsy.2022.01.008

93) Wang, R., Li, C., & Zhang, Y. (2023). End-to-end learning for flexible job shop scheduling with dual-attention reinforcement learning. Engineering Applications of Artificial Intelligence, 120, 105880. https://doi.org/10.1016/j.engappai.2022.105880

94) Wu, G., Pedrycz, W., Zhang, Z., & Ling, X. (2020). A particle swarm optimization with the surprisingly popular algorithm for large-scale optimization problems. *Information Sciences, 512,* 1044–1062. https://doi.org/10.1016/j.ins.2019.09.029

95) Wu, G., Pedrycz, W., Zhang, Z., & Ling, X. (2023). Hybrid PSO–SPA with adaptive Euclidean topology for large-scale heterogeneous optimization. *Applied Soft Computing, 135,* 110058. https://doi.org/10.1016/j.asoc.2023.110058

96) Xia, W., & Wu, Z. (2005). An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems. Computers & Industrial Engineering, 48(2), 409–425. https://doi.org/10.1016/j.cie.2005.01.018

97) Xing, L., Chen, Y., & Yang, K. (2010). A knowledge-based ant colony optimization for flexible job shop scheduling problems. Applied Soft Computing, 10(3), 888–896. https://doi.org/10.1016/j.asoc.2009.09.001

98) Xu, Y., et al., "Multi-agent reinforcement learning for intelligent manufacturing scheduling," Robotics and Computer-Integrated Manufacturing, 2025.

99) Yang, C., He, Y., Xu, X., & Wang, L. (2021). Artificial bee colony with adaptive covariance matrix for optimization problems. *Applied Soft Computing, 113,* 107930. https://doi.org/10.1016/j.asoc.2021.107930

100) Yanibelli, V., & Amandi, A. (2013). A multi-objective evolutionary algorithm combined with simulated annealing for scheduling problems. Expert Systems with Applications, 40(13), 5505–5512. https://doi.org/10.1016/j.eswa.2013.04.008

101) Yuan, Y., Xu, H., & Liu, J. (2013). A hybrid harmony search algorithm for the flexible job shop scheduling problem. Computers & Industrial Engineering, 64(3), 757–766. https://doi.org/10.1016/j.cie.2012.12.015

102) Zambrano Rey, G., Bermudez, J., & Ochoa, J. (2014). Semi-hierarchical optimization architecture for modern flexible manufacturing systems. International Journal of Computer Integrated Manufacturing, 27(6), 529–543. https://doi.org/10.1080/0951192X.2013.834477

103) Zheng, Y., Wang, S., & Wang, L. (2018). A two-dimensional tabu search for scheduling AGVs and machines in FMS. Computers & Operations Research, 98, 24–36. https://doi.org/10.1016/j.cor.2018.05.004

104) Zou, F., Li, J., & Zhang, Y. (2023). A self-adaptive iterated greedy algorithm for multi-AGV scheduling with maintenance constraints. Omega, 116, 102773. https://doi.org/10.1016/j.omega.2023.102773

105) Zhao, Q., et al., "Energy-aware flexible job shop scheduling considering AGV transportation," Journal of Cleaner Production, 2025.

106) Zhao, Y., Wang, S., "Constraint programming and hybrid methods for maintenance-aware FJSSP," Computers & Operations Research, 2025.

List of publications

SCI Journals:

1. Kaur Gaganpreet, R. S. Mishra and A. K. Madan "Effective Hybrid Novel Genetic and Adaptive Artificial Bee Colony Metaheuristic Algorithm for transforming concurrent Scheduling Problems" in International journal of Industrial Engineering, Theory , Applications and Practice.Vol. 31 No. 6 (2024) https://doi.org/10.23055/ijietap.2024.31.6.9709

2. Kaur Gaganpreet, R. S. Mishra and A. K. Madan "Enhancing Production Efficiency and Energy Conversation through Integrated Scheduling in Industry 4.0" in Journal of Sustainability ISSN 2071-1050

Conference

1. Published Paper In International Conference "Trajectory Generation Techniques – A Study" In ICAPIE-2016 Held At DTU
2. Published Paper In International Conference "An Advanced Hybrid Genetic Algorithm With Pareto Emphasis For Multi-Objective Scheduling Challenges "In 8th International Conference On Advanced Production And Industrial Engineering From August 28-31 2024
3. Published Paper In International Conference "Perto Based Metaheuristic Novel Hybrid Genetic Algorithm For Multi Objective Scheduling" At International Conference On Advancements And Key Challenges In Green Energy And Computing 24- 25 Feb 2023.

Non Scopus journals

1. Kaur Gaganpreet, R. S. Mishra And A. K. Madan "Metaheuristic Optimisation Algorithms Used In Flexible Manufacturing Techniques: A Review "In International Journal Of Research In Engineering And Innovation vol 3, Issue 4(2019),240-244

2. Kaur Gaganpreet, R. S. Mishra And A. K. Madan "Optimisation Of Scheduling Techniques In FMS In The Context Of Indian Industry "In The International Journal Of Research In Engineering And Innovation Volume -I Issue 6 (2017),47-56

3. Kaur Gaganpreet, R. S. Mishra And A. K. Madan "Scheduling Of AGV In FMS Using Metaheuristic Techniques "In International Journal Of Engineering And Techniques. Volume 8, Issue 4 August 2022.

4. Kaur Gaganpreet "Optimisation Of Scheduling In FMS Using Heuristic Approach: A Case Study" In International Journal Of Trends In scientific research.