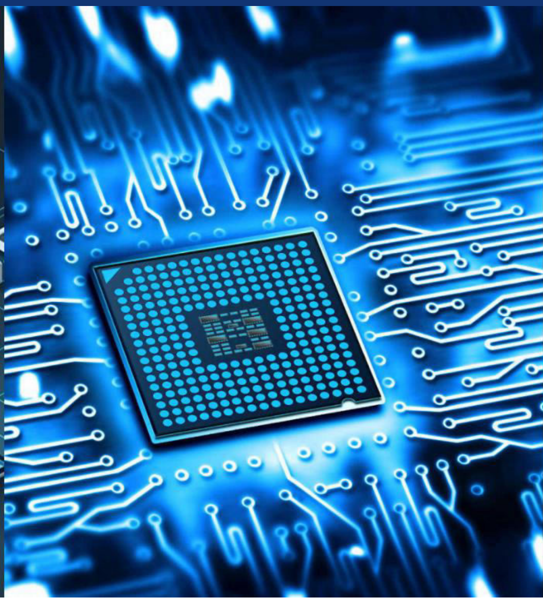
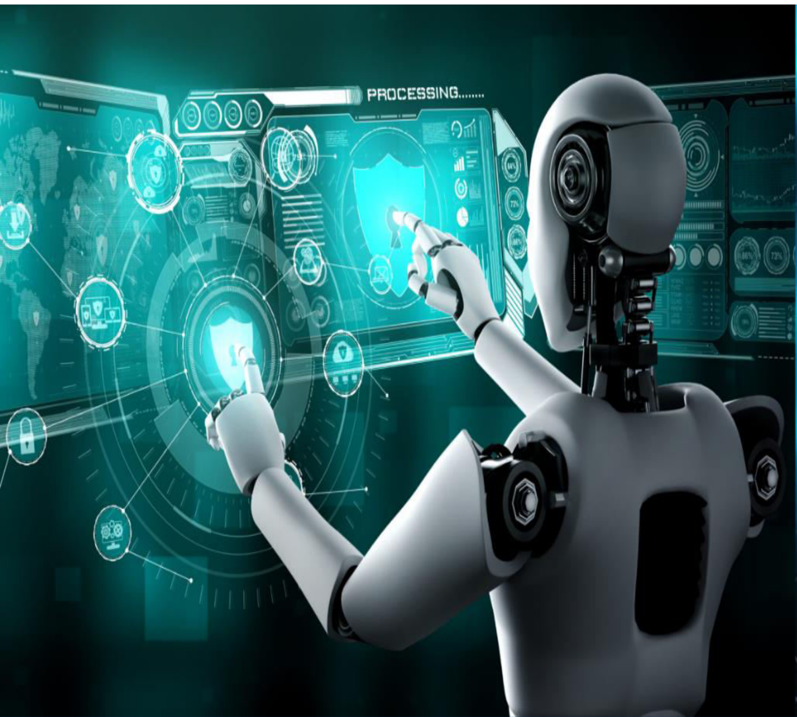
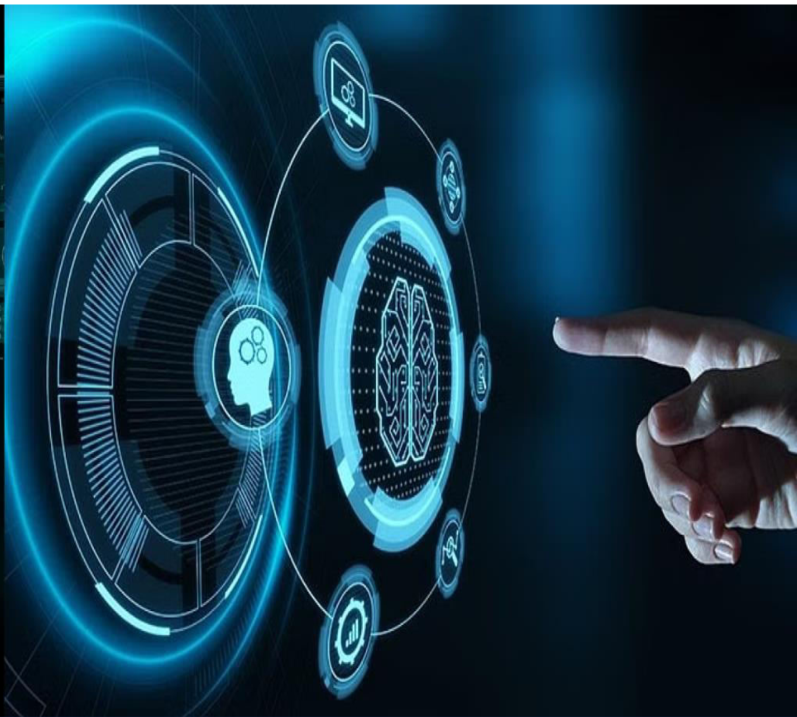
ISSN(O): 2320-9801
ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)





Impact Factor: 8.771

Volume 14, Issue 4, April 2026

www.ijircce.com ijircce@gmail.com +91-9940572462 +91 63819 07438

© 2026 IJIRCCE | Volume 14, Issue 4, April 2026 DOI: 10.15680/IJIRCCE.2026.1404101

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Enhancing Support Vector Machine Classification Using a Metaheuristic Optimization Approach

Venchiryala Nitya[1], Bandela Thirumala[2], Mrs. Vedavathi.K[3], Dr.V. Subbaramaiah[4], Dr.K. Rajitha[5]

Student, Department of Computer Science and Engineering, Mahatma Gandhi Institute of Technology, Gandipet, India[1,2]

Assistant Professor, Department of Computer Science and Engineering, Mahatma Gandhi Institute of Technology, Gandipet, India[3,4,5]

ABSTRACT: The Flexible Job Shop Scheduling Problem (FJSP) is an NP-hard optimization problem in which each operation can be processed on one of several eligible machines, making scheduling highly complex in modern manufacturing systems. This paper presents a comparative study between a Support Vector Machine (SVM)-based scheduling approach and the Non-dominated Sorting Genetic Algorithm II (NSGA-II) for optimizing three key objectives: minimizing makespan, minimizing average waiting time, and minimizing total waiting time. The SVM-based model is used to support scheduling decisions through predictive prioritization, while NSGA-II performs population-based multi-objective search to generate Pareto-optimal schedules. Experimental analysis on benchmark-style instances indicates that NSGA-II generally provides stronger global optimization and better objective trade-offs, whereas SVM-based scheduling offers faster computational response and practical usefulness in near real-time environments. The findings suggest that both methods are valuable under different operational conditions, and that hybrid strategies can further improve performance in flexible scheduling contexts.

KEYWORDS: Flexible Job Shop Scheduling, Support Vector Machine, NSGA-II, Multi-objective Optimization, Makespan, Waiting Time, Pareto Optimization.

I. INTRODUCTION

Flexible Job Shop Scheduling has become increasingly important due to the demand for adaptive and efficient production systems where alternative machine routing is available for each operation. Unlike classical job shop scheduling, FJSP allows each task to be assigned to one among multiple machines, which increases flexibility but also significantly expands the search space. In practice, industries do not optimize a single objective only; they must balance productivity and flow efficiency by reducing completion time and idle delays. This motivates multi-objective optimization where makespan, average waiting time, and total waiting time are optimized simultaneously. NSGA-II is a well-established evolutionary method for handling conflicting objectives through Pareto-front generation, while SVM provides a machine learning perspective that can quickly predict beneficial scheduling priorities from extracted shop-floor features. This paper compares these two approaches under a common framework to analyze solution quality, computational efficiency, and operational suitability.

II. LITERATURE SURVEY

[1] An effective multi-objective metaheuristic for the support vector machine with feature selection by Mathias Badilla-Salamanca and Rosa Medina Durán (2025, IEEE) proposes a metaheuristic-based Support Vector Machine model optimized for feature selection. The method improves classification accuracy across various binary datasets by fine- tuning SVM parameters through multi-objective optimization. The approach shows superior performance and flexibility in handling different data types. However, the method suffers from high computational cost, especially when applied to large datasets.

[2]A Survey Study on Metaheuristic-based Feature Selection Approaches of Intrusion Detection Systems in Distributed Networks by Yashar, Misaeid, and Javid (2025) provides a broad review of metaheuristic-based feature selection techniques used in intrusion detection systems. The survey highlights efficiency in improving detection accuracy and feature reduction. Existing datasets are identified as strong benchmarks for future studies. However, the models still face challenges in execution speed and generalization to unseen attacks.

[3] A disjunctive graph-based metaheuristic for flexible job-shop scheduling problems considering fixture shortages in customized manufacturing systems by J. Li et al. (2025) introduces a genetic algorithm integrated with disjunctive graphs to address fixture shortages in flexible job-shop environments. The method schedules jobs efficiently while accounting for limited resources and machine constraints. Overall scheduling performance improves, but the approach suffers from high computational complexity, making scalability difficult for large systems.

[4] Hybrid multi-objective optimization with NSGA-II for FS by P. Vijai (2025) presents a hybrid optimization model that combines NSGA-II with additional learning techniques to enhance feature selection. Accuracy and convergence improve in multi-objective optimization settings. However, the resulting model becomes complex, requires careful tuning, and consumes more computational resources.

III. PROBLEM DEFINITION

The problem considered in this study is a multi-objective FJSP in which a set of jobs is composed of ordered operations and each operation has a predefined subset of eligible machines with machine-dependent processing times. The schedule must satisfy precedence constraints between operations of the same job, machine-capacity constraints that allow only one operation at a time on a machine, and non-preemptive execution rules once an operation starts. The optimization targets are defined as minimizing the maximum completion time across all jobs (makespan), minimizing the mean waiting time of operations in queues, and minimizing the cumulative waiting time across the entire schedule. Since these objectives are often conflicting, the goal is not a single universal optimum but a set of high-quality trade-off solutions.

IV. PROPOSED SYSTEM

The proposed system introduces a comparative architecture where SVM and NSGA-II are implemented over the same FJSP instances and evaluated using identical objective metrics. In the SVM-based branch, relevant state features such as machine utilization, queue lengths, operation remaining time, and job urgency are used to train the model for priority prediction that supports operation-machine assignment decisions. In the NSGA-II branch, chromosome encoding represents operation sequence and machine allocation, and evolutionary operators such as crossover and mutation are applied with non-dominated sorting and crowding-distance preservation to evolve diverse Pareto solutions. Both branches produce schedules that are then assessed using makespan, average waiting time, and total waiting time, enabling a fair and structured comparison.

V. EXISTING SYSTEM

Existing scheduling systems in many studies and manufacturing settings rely on dispatching rules, heuristic methods, or single-objective optimization frameworks. Rule-based approaches such as shortest processing time are computationally cheap but often fail to produce globally competitive solutions when machine flexibility and objective conflicts increase. Classical metaheuristics improve exploration but may be tuned mainly for makespan and may not adequately reflect waiting-time-related performance. In dynamic production environments, methods that require high computational time can also become difficult to deploy for frequent rescheduling. These limitations highlight the need for comparative evaluation of faster learning-based approaches and stronger multi-objective evolutionary methods.

VI. DESIGN AND METHODOLOGY

The design follows a pipeline in which FJSP instance data first passes through preprocessing and feature extraction, after which two parallel optimization routes are executed: predictive scheduling through SVM and evolutionary optimization through NSGA-II. The outputs from both routes are transformed into executable schedules and then

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

evaluated through objective computation and comparative statistical analysis. Conceptually, the system diagram begins with input instances, continues to preprocessing, splits into SVM and NSGA-II modules, merges at schedule evaluation, and concludes at result analytics and visualization. From an object-oriented perspective, the class-level design includes core entities such as Job, Operation, Machine, and Schedule, with a common Scheduler abstraction that is implemented by SVMScheduler and NSGA2Scheduler subclasses.

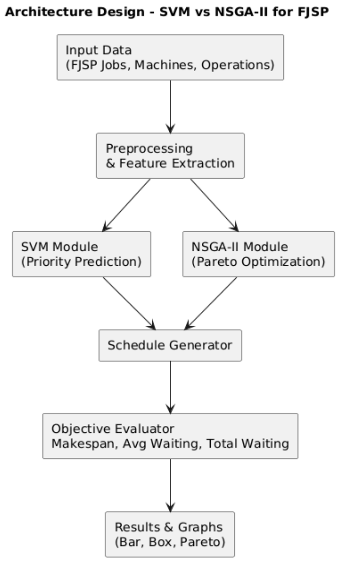
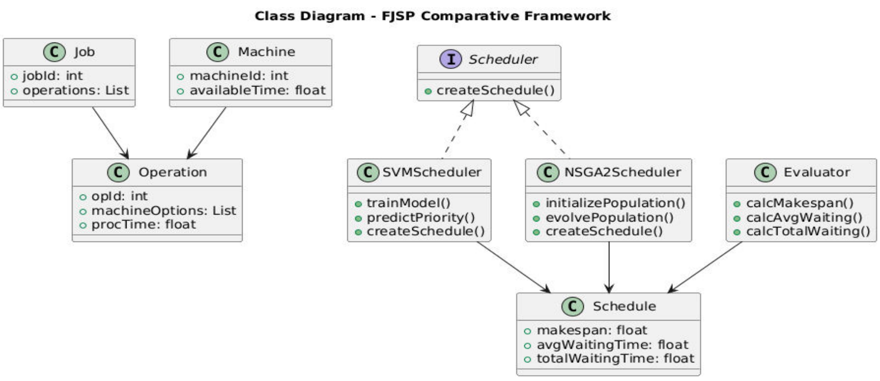


Figure-1 shows the **architecture diagram** shows the complete workflow of your system in a simple way. First, FJSP input data (jobs, operations, machine choices, processing times) is collected and preprocessed. After preprocessing, the flow is split into two parallel modules: the **SVM module**, which predicts operation priority for faster scheduling decisions, and the **NSGA-II module**, which searches for high-quality Pareto-optimal schedules by evolutionary optimization. Outputs from both modules go to a common **schedule generator**, which builds executable schedules. These schedules are then passed to the **objective evaluator**, where makespan, average waiting time, and total waiting time are computed. Finally, the evaluated outputs are shown in the **results and graphs** block for comparison using plots like bar charts, box plots, and Pareto scatter plots. So, this diagram mainly explains *how data moves* through your comparative system.



The **class diagram** explains the static structure of your implementation, meaning which classes exist and how they relate.Job,Operation and Machine are core problem entities. A Job contains multiple Operation objects, and each operation can be assigned to eligible machines. Schedular stores final performance values such as makespan and waiting times.Schedular is a common interface so both algorithms follow the same scheduling contract(createSchedular()), making comparison fair.SVMShedular implements learning-based scheduling through training and prediction methods, while NSGA2Schedular implements population initialization and evolution methods for multi-objective search.Evaluator is responsible for computing the three objective values from generated schedules. In short, this diagram explains *how the code is organized* and how both approaches fit into one comparable framework.

VII. IMPLEMENTATION

7.1 Development Environment

The proposed comparative framework was implemented in Python because it provides strong support for optimization, machine learning, and scientific visualization in a single ecosystem. The implementation was executed on a standard workstation configuration with multi-run experimentation enabled to reduce randomness effects in performance comparison.

7.2 Data Preparation Module

The first implementation stage focused on building a clean input pipeline for Flexible Job Shop Scheduling instances. Each instance file was parsed into structured objects representing jobs, operations, machine eligibility sets, and machine-specific processing times. After parsing, preprocessing routines validated precedence constraints, removed malformed records, and standardized indexing so both SVM and NSGA-II modules could consume identical input representations. This design decision was important for fairness because any inconsistency in input formatting would bias the comparative analysis.

7.3 Core Scheduling Representation

A shared schedule representation was created for both algorithms. In this representation, each schedule contains machine assignments, operation start times, completion times, and derived metrics. Using one common schedule object avoided duplication and made evaluation consistent. This module is intentionally simple but central to the whole system.

7.4 SVM-Based Scheduler Implementation

The SVM branch was implemented as a predictive priority model. Feature vectors were generated from current shop-floor states, including queue size, machine utilization, remaining processing workload, and operation-level urgency. The trained SVM model predicts a priority score, and operations with better scores are dispatched earlier under feasibility constraints. The SVM implementation emphasizes fast inference, so it is suitable for scenarios where the scheduler must react quickly.

7.5 NSGA-II Implementation

The NSGA-II branch was implemented with chromosome encoding that combines operation sequence information and machine assignment choices. Initial populations were generated randomly within feasibility limits, then improved over generations through crossover and mutation operators. Non-dominated sorting and crowding-distance logic were used to preserve a diverse set of high-quality solutions rather than forcing a single-objective optimum.Compared to SVM, this module is computationally heavier, but it provides better multi-objective exploration in complex FJSP instances.

7.6 Objective Evaluation Engine

A separate evaluator module was implemented to compute makespan, average waiting time, and total waiting time from any generated schedule. This evaluator is shared by both branches and is called after every run. Because all metrics are measured through the same logic, the comparison remains methodologically sound.The evaluator also logs runtime and stores each run’s results for later statistical analysis.

7.7 Results Logging and Visualization

All trial outputs were exported into tabular result files and then used for graph generation. Visualization scripts produced bar charts for mean comparison, box plots for stability analysis, convergence curves for optimization behavior, and Pareto scatter plots for trade-off quality inspection. These plots were integrated into the testing-and-results section of the study to make algorithm differences interpretable beyond raw numeric tables.

© 2026 IJIRCCE | Volume 14, Issue 4, April 2026 DOI: 10.15680/IJIRCCE.2026.1404101

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

VIII.TESTING AND RESULTS

8.1 Convergence Analysis on Brandimarte Instances

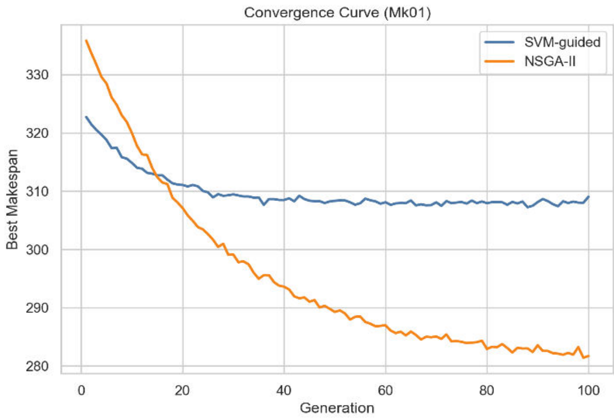


Figure 3:Shows the Mk01 convergence plot shows SVM improving quickly at the beginning but flattening early. NSGA-II improves more steadily across generations and reaches a lower final makespan. This indicates better long-run optimization capability for NSGA-II.

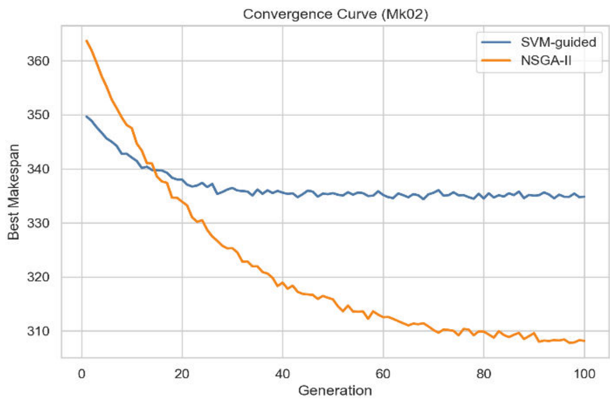


Figure 4:Shows Mk02, the same trend appears, with SVM converging faster initially but stopping at a makespan. NSGA-II continues refining solutions for longer and ends with better best-makespan values.

8.2 Summary of Numerical Results from Brandimarte CSV

brandimarte_mk01_mk02_summary

Instance	Best Makespan (SVM)	Best Makespan (NSGA-II)	Avg Waiting (SVM)	Avg Waiting (NSGA-II)	Runtime sec (SVM)	Runtime sec (NSGA-II)
Mk01	289.95	267.86	30.36	26.12	2.7	14.8
Mk02	313.52	286.47	33.14	29.06	2.9	15.6

8.2 Summary of Numerical Results from Brandimarte CSV

Figure 5:Shows the barndimarte mk01,mk02 csv file numerically summarizes both instances and confirms the visual trends from the plots. NSGA-II achieves better objective values, while SVM has lower runtime. Together, they show the quality-versus-speed trade-off clearly.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

IX. CONCLUSION AND FUTURE SCOPE

This study demonstrates that both SVM and NSGA-II are meaningful approaches for multi-objective flexible job shop scheduling, but they serve different operational priorities. NSGA-II is more effective for obtaining high-quality global trade-offs among makespan and waiting-time objectives, whereas SVM is advantageous for faster decision support in time-sensitive environments. The comparative evaluation confirms that algorithm suitability depends on whether the scheduling context prioritizes optimization depth or response speed. Future work can focus on hybrid frameworks in which SVM guides the initialization or search direction of NSGA-II to combine predictive speed with evolutionary optimization strength. The model can also be extended to dynamic FJSP scenarios involving machine breakdowns, urgent job arrivals, and stochastic processing times. Additional objectives such as energy consumption, tardiness penalties, and setup costs can be integrated to better reflect industrial realities. Finally, validation on real factory datasets and digital twin environments can improve generalizability and practical adoption of the proposed comparative framework.

REFERENCES

[1] M.Badilla-Salamanca, R. Medina, and C. Contreras-Bolton, "An effective multi- objective metaheuristic for the support vector machine with feature selection," *Knowledge-Based Systems*, vol.328, p.114203,2025. DOI: 10.1016/j.knosys.2024.114203

[2] Y.Pourardebilkhah, M.Hosseini Shirvani, and J.Taheri, "A survey study on meta-heuristic-based feature selection approaches of intrusion detection systems in distributed networks," *Computers & Standards & Interfaces*,vol.96,p.104074,2025. DOI: 10.1016/j.csi.2025.104074

[3] J.Li,"A disjunctive graph-based metaheuristic for flexible job-shop scheduling with fixture shortages," FlexibleServices&Manufacturing,Journal,vol.37,pp.1-23,2025. DOI: 10.1007/s10696-025-0353

[4] P.Vijai, "A hybrid multi-objective optimization approach with NSGA-II and hybrid filter methods for feature_selection,"*Journal_of_Intelligent&Fuzzy-Systems*,vol.49,pp.1-18,2025. DOI: 10.1016/j.jifs.2025.000062

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com