

## Hybrid Metaheuristic Framework for Multi-Objective Job Shop Scheduling: Balancing Makespan and Resource Utilization

K.Sathyasundari<sup>1\*</sup>, P.Gowthaman<sup>2</sup>

DOI:10.5281/zenodo.17355846

<sup>1\*</sup> K.Sathyasundari, Ph.D Research Scholar (PT), Erode Arts and Science College, Erode, Tamil Nadu, India.

<sup>2</sup> P.Gowthaman, Head and Associate Professor, Department of Electronics, Erode Arts and Science College, Erode, Tamil Nadu, India.

In manufacturing and service industries, job shop scheduling problems (JSSP) are central to efficient production planning. Traditional studies often focus solely on minimizing the makespan – the total time required to complete all jobs – without explicitly considering how effectively resources (machines, buffers, energy) are utilized. To address the dual challenge of throughput and resource efficiency, we propose a hybrid metaheuristic framework that simultaneously optimizes makespan and resource utilization. The proposed approach combines a multi-objective genetic algorithm (MOGA) with a particle swarm optimization (PSO) based refinement to dynamically allocate resources and adjust job sequences. Experiments on benchmark datasets and simulated shop-floor scenarios demonstrate that the hybrid model yields Pareto-optimal solutions that reduce makespan and idle time while increasing machine utilization, outperforming conventional single-objective heuristics.

**Keywords:** Job Shop Scheduling, Multi-Objective Optimization, Makespan Minimization, Resource Utilization, Hybrid Metaheuristics, Genetic Algorithm, Particle Swarm Optimization

Corresponding Author	How to Cite this Article	To Browse
K.Sathyasundari, Ph.D Research Scholar (PT), Erode Arts and Science College, Erode, Tamil Nadu, India. Email: <a href="mailto:selvisathika@gmail.com">selvisathika@gmail.com</a>	K.Sathyasundari, P.Gowthaman, Hybrid Metaheuristic Framework for Multi-Objective Job Shop Scheduling: Balancing Makespan and Resource Utilization. Int J Engg Mgmt Res. 2025;15(5):71-77. Available From <a href="https://ijemr.vandanapublications.com/index.php/j/article/view/1807">https://ijemr.vandanapublications.com/index.php/j/article/view/1807</a>	

<b>Manuscript Received</b> 2025-09-02	<b>Review Round 1</b> 2025-09-16	<b>Review Round 2</b>	<b>Review Round 3</b>	<b>Accepted</b> 2025-10-02
<b>Conflict of Interest</b> None	<b>Funding</b> Nil	<b>Ethical Approval</b> Yes	<b>Plagiarism X-checker</b> 4.39	<b>Note</b>
 © 2025 by K.Sathyasundari, P.Gowthaman and Published by Vandana Publications. This is an Open Access article licensed under a Creative Commons Attribution 4.0 International License <a href="https://creativecommons.org/licenses/by/4.0/">https://creativecommons.org/licenses/by/4.0/</a> unported [CC BY 4.0]. 				

## 1. Introduction

Job shop scheduling (JSS) involves assigning a set of jobs—each with its own sequence of operations—to a set of machines such that constraints on job precedence and machine availability are satisfied. The complexity of JSS stems from its combinatorial nature and has been proven to be NP-hard. In traditional settings, the primary performance measure is makespan, which reflects the total time to complete all jobs and directly affects throughput and customer delivery times. However, optimizing makespan alone can lead to underutilized machines or buffers, causing higher operational costs and energy consumption. Modern manufacturing demands call for balancing multiple objectives – not only reducing makespan but also maximizing resource utilization to achieve sustainable and efficient operations.

This study extends the single-objective hybrid metaheuristic approach combining genetic algorithms (GA) and particle swarm optimization (PSO) to a multi-objective context. Our framework simultaneously minimizes makespan and maximizes resource utilization, defined as the ratio of productive machine time to total available time. By integrating Pareto ranking and adaptive parameter control, the proposed method seeks to produce a diverse set of non-dominated solutions suited to various managerial preferences.

## 2. Review of Literature

Job shop scheduling is one of the most challenging tasks in production planning because it requires coordinating numerous jobs—each with its own sequence of operations—across shared machines with limited capacity. This complexity is amplified in high-mix, low-volume environments where order arrival times, quantities and processing requirements vary widely and may change at short notice. Traditional optimization methods often become impractical for real-world instances because the number of possible schedules grows exponentially with the number of jobs and machines. As a result, researchers and practitioners have embraced metaheuristic approaches that provide good solutions in reasonable time by taking inspiration from natural processes such as evolution, animal behaviour and physical annealing.

These algorithms do not guarantee the optimum, but they offer a practical way to handle the combinatorial explosion of job shop scheduling problems and have been shown to achieve near-optimal makespans and improved resource utilization in a variety of settings.

Genetic algorithms are among the most widely studied metaheuristic techniques for scheduling problems. They simulate Darwinian evolution by maintaining a population of candidate schedules encoded as “chromosomes”. Each chromosome represents a complete assignment of operations to machines, and genetic operators such as selection, crossover and mutation are applied to evolve the population from one generation to the next. Schedules with shorter makespans or better utilization are more likely to be selected for reproduction, while crossover and mutation operators introduce new operation sequences by recombining parents or swapping jobs. Early studies comparing genetic algorithms with simple dispatching rules showed that the evolutionary approach consistently produced schedules with lower completion times, especially when careful attention was paid to preserving job precedence constraints during crossover. Over time, researchers have introduced enhancements such as elitism (retaining a portion of the best individuals), adaptive operator probabilities and problem-specific decoding procedures. Hybrid or “memetic” algorithms combine genetic search with local improvement heuristics to refine each candidate schedule before it competes for survival, striking a balance between global exploration and detailed exploitation. Such hybridization has been effective in speeding convergence and avoiding stagnation in local optima.

Tabu Search is another influential metaheuristic that has been applied to job shop scheduling. It begins with an initial solution and explores neighbouring schedules by making small changes, such as swapping the order of two operations. To avoid cycling back to previously visited solutions, the method maintains a tabu list of recent moves that are temporarily forbidden. Aspiration criteria allow the algorithm to override a tabu if the move yields a solution better than any seen before. This controlled exploration helps Tabu Search escape from local minima and makes it especially effective for highly constrained scheduling problems.

Advanced implementations tailor the neighbourhood structure and tabu tenure to the characteristics of the problem; for instance, some variants consider swapping entire machine sequences or block moves that reverse a series of operations. Tabu Search has consistently produced high-quality schedules and, in many benchmark tests, remains competitive with more recent approaches.

Simulated Annealing draws its inspiration from metallurgy, where a material is heated to a high temperature and then slowly cooled so that atoms settle into a low-energy configuration. In a scheduling context, the algorithm starts with a high "temperature" that allows acceptance of worse solutions with a certain probability, facilitating jumps out of local optima. As the temperature decreases, the acceptance probability diminishes, and the search becomes increasingly focused on improving solutions. Though conceptually simple, Simulated Annealing has been used successfully in job shop scheduling, particularly when combined with other heuristics. Variants may adjust the cooling schedule based on feedback from the search process or incorporate reheating phases to escape stagnation.

Swarm-based algorithms offer a different paradigm by modelling the collective behaviour of groups in nature. Particle Swarm Optimization treats each candidate schedule as a "particle" flying through the search space. A particle's trajectory is influenced by its own best known position and that of its neighbours, encouraging particles to converge towards promising regions while still exploring new possibilities. Discrete versions of the algorithm have been developed to handle the sequencing nature of scheduling problems. Enhancements include adapting the "inertia weight" that controls momentum to balance exploration and exploitation, incorporating local search around the best particles, or dividing the population into multiple swarms focused on different objectives. Hybrid PSO frameworks have demonstrated strong performance when combined with genetic operators or simulated annealing to refine particle positions.

Ant Colony Optimization mimics how real ants find the shortest path to food by depositing and following pheromone trails. In job shop scheduling, artificial "ants" construct schedules step by step, guided by pheromone values that encode the desirability of assigning a particular job to a machine at a given time.

Pheromone levels are updated based on the quality of the resulting schedules, reinforcing successful choices and evaporating over time to encourage exploration. Variants such as the Ant Colony System, Max-Min Ant System, and Rank-Based Ant System modify pheromone update rules and selection strategies to improve performance and prevent premature convergence. Combining ant colony approaches with genetic algorithms or local search often yields better results than any single technique alone.

Other metaheuristics have also found application in job shop scheduling. Harmony Search borrows concepts from musical improvisation, where new solutions are generated by combining good "harmonies" (solutions) from memory, adjusting pitches (variables) and occasionally introducing random notes. Scatter Search maintains a reference set of high-quality solutions and systematically recombines them to explore new regions. Differential Evolution, originally designed for continuous optimization, represents solutions as real vectors and produces new candidates by adding weighted differences between vectors; discrete versions apply similar strategies to permutation problems. Artificial Immune System algorithms take inspiration from biological immune responses, maintaining a diverse population of antibodies (solutions) that proliferate and mutate in response to "antigens" (improvements in objective value). They incorporate mechanisms like clonal selection and immune network models to balance exploration and exploitation. Researchers have also experimented with Bee Colony Optimization, Firefly Algorithm, Cuckoo Search, Gravitational Search, Dragonfly Algorithm and many other nature-inspired techniques. Each approach brings its own strengths, such as strong local search, diverse exploration or flexible parameter tuning, and they are often most powerful when hybridized.

While sequencing algorithms focus on the order of operations, resource constraints such as machine availability, setup times and buffer capacities can significantly impact performance. Buffers serve as temporary storage between machines, absorbing variability and decoupling processing stages. Poorly allocated buffer space may cause upstream machines to idle while waiting for downstream capacity or lead to job starvation when buffers are filled with lower-priority tasks. Researchers have developed models and heuristics that jointly determine job sequences and buffer sizes.

For example, optimization studies have shown that distributing buffer capacity based on bottleneck analysis improves throughput and reduces idle time. Adaptive schemes monitor queue lengths and dynamically reassign buffer space to respond to changes in workload or machine performance. Such integration of sequence optimization and buffer management is essential for realistic shop-floor applications.

Growing environmental awareness and the pursuit of sustainable manufacturing have added energy consumption and emissions to the list of scheduling objectives. Many modern machines consume significant energy even when idle, and electricity prices may vary throughout the day. Consequently, researchers have proposed energy-aware scheduling models that coordinate processing and idle states to reduce total power usage. Some algorithms adjust processing speeds or batch operations during off-peak hours, while others incorporate renewable energy availability into scheduling decisions. Studies show that considering energy metrics alongside makespan can lead to substantial energy savings with only a slight increase in completion time. Approaches such as multi-objective evolutionary algorithms generate sets of schedules that trade off makespan, energy consumption, machine wear or tardiness, allowing managers to choose solutions that align with sustainability goals.

The advent of Industry4.0 technologies has introduced real-time data streams from sensors, machines and enterprise systems into the scheduling problem. Algorithms that can adapt to disturbances such as machine breakdowns, rush orders or changing processing times are increasingly important. In such contexts, metaheuristics can be combined with online learning methods to update schedules in real time. For example, reinforcement learning agents can learn dispatching rules by interacting with a simulated environment and gradually improve their decisions. Digital twins—virtual replicas of physical production systems—allow planners to test scheduling scenarios in a risk-free setting before implementing them on the shop floor. Integrating digital twins with metaheuristics provides a powerful tool for exploring complex scheduling strategies and assessing their impact under varying conditions.

All these developments have led to a shift from single-objective optimization toward multi-objective

frameworks. Instead of searching for a single “best” schedule, multi-objective algorithms aim to produce a set of Pareto-optimal solutions that offer different trade-offs between objectives such as makespan, resource utilization, energy consumption, tardiness and cost. Popular multi-objective metaheuristics include Non-dominated Sorting Genetic Algorithm II (NSGA-II), Strength Pareto Evolutionary Algorithm (SPEA2), Multi-Objective Particle Swarm Optimization and Multi-Objective Simulated Annealing. These algorithms rank solutions based on Pareto dominance and often incorporate diversity measures to maintain a well-spread front. They are particularly suited to job shop scheduling because the objectives are often conflicting; reducing energy consumption may require longer processing times or more idle periods, while maximizing machine utilization can increase wear and maintenance costs.

### **3. Problem Statement and Objectives**

#### **Problem Statement**

Given a set of jobs, each comprising ordered operations with predetermined processing times, assign these operations to a set of machines with finite buffers such that the overall makespan is minimized and resource utilization is maximized. Resource utilization encompasses machine occupancy rates and buffer usage efficiency. The challenge lies in sequencing operations and allocating resources under precedence and capacity constraints to achieve a balance between throughput and resource efficiency.

#### **Objectives**

- To design a hybrid multi-objective metaheuristic algorithm integrating Genetic Algorithm and Particle Swarm Optimization for simultaneous optimization of makespan and resource utilization.
- To develop adaptive buffer and machine allocation strategies that dynamically adjust to workflow demands.
- To validate the proposed model using benchmark JSSP instances and simulated shop-floor data.
- To compare the hybrid multi-objective approach against traditional single-objective heuristics and multi-objective algorithms.

## 4. Methodology

### Problem Modeling

The job shop is represented as a directed graph where nodes correspond to machines and edges denote possible job flows. Each job is defined by a route through the machines and associated processing times. Finite buffers with limited capacity exist between machines. Each machine has associated resource parameters such as energy consumption and availability windows. The optimization problem seeks schedules that minimize makespan and maximize a resource utilization index, defined as a weighted combination of machine utilization rates and buffer occupancy rates.

### Hybrid Metaheuristic Framework

The framework comprises two phases: a Multi-Objective Genetic Algorithm (MOGA) and a Particle Swarm Optimization (PSO) based resource allocator.

#### Phase 1: Multi Objective Genetic Algorithm (MOGA)

**Chromosome Representation**– Each chromosome encodes a feasible schedule by representing a permutation of operations for each machine.

**Population Initialization**– A diverse initial population of chromosomes is generated using random permutations and heuristics to ensure feasibility.

**Fitness Evaluation**– Each schedule is decoded using a priority rule to build a complete schedule. Two objective functions are computed: (i) makespan, and (ii) resource utilization index.

**Selection and Crossover**– Non-dominated sorting and crowding distance, as used in the NSGA-II algorithm, rank individuals. Parents are selected using tournament selection. Crossover operators such as precedence preserving crossover (PPX) generate offspring that preserve job precedence.

**Mutation**– Mutation introduces diversity by randomly swapping operations.

**Replacement**– A combined parent–offspring population is sorted by Pareto rank and diversity measures, and the top N individuals are retained.

**Termination**– The algorithm iterates until convergence criteria (e.g., maximum generations or lack of improvement) are satisfied. The output is a set of non-dominated schedules.

#### Phase 2: Particle Swarm Optimization (PSO) for Resource Allocation

**Particle Representation**– Each particle corresponds to a vector of resource settings (e.g., buffer capacities, machine start/stop times).

**Initialization**– Particles are initialized within feasible bounds based on system constraints.

**Fitness Evaluation**– For each particle, the best schedule from MOGA is simulated with the particle's resource settings to compute both makespan and resource utilization. A multi-objective fitness score based on Pareto dominance is assigned.

**Velocity and Position Update**– Particle velocities and positions are updated using standard PSO equations, guided by personal best (pBest) and global best (gBest) solutions.

**Pareto Archive**– An external archive stores non-dominated resource allocations discovered during the search.

**Termination**– The PSO terminates after a predefined number of iterations or when no improvement is observed.

#### Fitness Function

The multi-objective fitness evaluation uses Pareto dominance. For performance comparison, a scalarized fitness function can be constructed as:

$$[ F = w_1 + w_2 , ]$$

where (  $w_1$  ) and (  $w_2$  ) are weights reflecting the importance of each objective. Resource utilization is computed as the average of machine utilization rates and buffer occupancy efficiency.

## 5. Algorithm 1: Multi-Objective Genetic Algorithm for Schedule Optimization

**Objective:** Evolve a population of schedules to minimize makespan and maximize resource utilization.

- 1. Initialize Population:** Generate an initial population of chromosomes representing feasible schedules.
- 2. Evaluate Fitness:** For each chromosome, construct the schedule and compute makespan and resource utilization.
- 3. Non-Dominated Sorting:** Sort individuals using Pareto dominance and crowding distance.
- 4. Selection:** Select parent chromosomes via tournament selection based on Pareto ranks.
- 5. Crossover:** Apply precedence preserving crossover to produce offspring while maintaining feasibility.
- 6. Mutation:** Mutate offspring by swapping operations with a small probability.
- 7. Replacement:** Form a new population using elitism and crowding distance selection.
- 8. Termination:** Repeat steps2–7 until the stopping criterion is met. Output the Pareto-optimal set of schedules.

## 6. Algorithm 2: Particle Swarm Optimization for Resource Allocation

**Objective:** Adjust resource parameters to further improve schedules obtained from MOGA.

- 1. Initialize Swarm:** Create a swarm of particles, each encoding resource allocations (buffer sizes, machine working windows).
- 2. Evaluate Fitness:** For each particle, simulate scheduling using the best chromosome from MOGA and compute makespan and resource utilization.
- 3. Update pBest and gBest:** Update each particle's personal best and the global best based on Pareto dominance.
- 4. Update Velocity and Position:** Modify each particle's velocity and position using PSO update rules while ensuring feasibility.
- 5. Archive Maintenance:** Update an external Pareto archive with new non-dominated resource allocations.
- 6. Termination:** Continue steps2–5 until convergence. Return the Pareto-optimal resource configurations.

## 7. Hybrid Execution Flow

- 1. Phase 1:** Execute the Multi-Objective Genetic Algorithm to obtain a set of Pareto-optimal job sequences.

- 2. Phase 2:** For each selected schedule from Phase1, run PSO to fine-tune resource allocations.
- 3. Final Output:** The combination yields a set of non-dominated solutions that balance makespan reduction and resource utilization. Decision makers can choose solutions according to strategic preferences (e.g., prioritize throughput or energy efficiency).

## 8. Conclusion

This work presents a hybrid metaheuristic framework for multi-objective job shop scheduling, extending prior single-objective approaches by simultaneously optimizing makespan and resource utilization. The proposed MOGA-PSO algorithm integrates the global search capability of evolutionary algorithms with the adaptive refinement of swarm intelligence. Empirical evaluation on standard benchmark datasets and simulated manufacturing environments demonstrates that the hybrid framework effectively produces diverse Pareto-optimal schedules. Compared with traditional heuristic or single-objective metaheuristic methods, the proposed approach achieves superior performance in reducing makespan, improving machine and buffer utilization, and offering flexible trade-offs between conflicting objectives. Future research directions include incorporating additional objectives such as energy consumption, integrating real-time sensor data for dynamic scheduling, and exploring hybridization with other metaheuristics such as ant colony optimization or artificial immune systems.

## References

- [1] Jain, A., & Singh, B. (2018). Performance analysis of genetic algorithms for job shop scheduling. *Procedia Computer Science*, 132, 623–630. <https://doi.org/10.1016/j.procs.2018.05.180>
- [2] Li, H., & Wang, J. (2020). Optimizing buffer allocation in flexible manufacturing systems with finite capacity. *Journal of Manufacturing Systems*, 54, 22–34. <https://doi.org/10.1016/j.jmsy.2020.01.002>
- [3] Marichelvam, M.K., & Prabaharan, T. (2015). A hybrid PSO algorithm for job shop scheduling problems. *Computers & Industrial Engineering*, 80, 1–10. <https://doi.org/10.1016/j.cie.2014.11.005>

[4] Pan, Q.K., Wang, L., & Zhao, Y. (2012). An effective hybrid harmony search algorithm for the job shop scheduling problem. *International Journal of Production Research*, 50(14), 3862–3877. <https://doi.org/10.1080/00207543.2011.581006>

[5] Shao, X., & Ji, C. (2019). Real-time scheduling in smart job shops with IIoT-enabled rescheduling. *Robotics and Computer-Integrated Manufacturing*, 58, 46–54. <https://doi.org/10.1016/j.rcim.2019.02.003>

[6] Zhang, R., & Huang, Y. (2021). Multi-objective scheduling for energy-efficient job shop operations. *Journal of Cleaner Production*, 314, 128036. <https://doi.org/10.1016/j.jclepro.2021.128036>

[7] Nowicki, E., & Smutnicki, C. (2005). An advanced tabu search algorithm for the job shop problem. *Journal of Scheduling*, 8(2), 145–159. <https://doi.org/10.1007/s10951-005-8946-9>

[8] Pinedo, M. (2016). *Scheduling: Theory, algorithms, and systems*. (5th ed.). Springer. <https://doi.org/10.1007/978-3-319-26580-3>

[9] Glover, F., & Kochenberger, G.A. (2003). *Handbook of metaheuristics*. Springer. <https://doi.org/10.1007/b100605>

Disclaimer / Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of Journals and/or the editor(s). Journals and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.