

Research Proposal

Topic: Decentralized and Robust Scheduling for Flexible Job Shops using Multi-Agent Reinforcement Learning (MARL)

Student Name: Adarsh

Department: Mechanical Engineering, Delhi Technological University (DTU)

1. Introduction

The manufacturing industry is currently undergoing a paradigm shift toward **Industry 4.0**, characterized by the integration of cyber-physical systems and data-driven decision-making. At the core of high-mix, low-volume production lies the **Flexible Job Shop Scheduling Problem (FJSSP)**. Unlike traditional flow shops, flexible job shops allow operations to be performed on multiple capable machines, creating a complex optimization challenge.

Traditional scheduling methods (e.g., Linear Programming, static Gantt charts) rely on centralized decision-making that assumes a deterministic environment. However, real-world shop floors are stochastic; machine breakdowns, tool wear, and rush orders render static schedules obsolete within minutes. This research proposes moving from a **Centralized/Static** paradigm to a **Decentralized/Dynamic** paradigm using **Multi-Agent Reinforcement Learning (MARL)**.

2. Problem Statement

Current Computer-Aided Manufacturing (CAM) scheduling systems suffer from three critical limitations:

1. **Fragility:** A single machine breakdown invalidates the entire global schedule, requiring computationally expensive rescheduling (the "nervousness" of the system).
2. **Latency:** Centralized algorithms (like Genetic Algorithms) struggle to react in real-time to micro-disturbances on the shop floor.
3. **Single-Point of Failure:** A central scheduler acts as a bottleneck; if the server fails or lags, the entire production line stalls.

There is a critical need for a **"Self-Healing" scheduling architecture** where machines can autonomously negotiate and re-allocate tasks in response to failure, minimizing idle time without human intervention.

3. Research Objectives

The primary objective is to develop a robust scheduling framework that minimizes **Makespan** (total production time) while maximizing **Robustness** (stability under failure).

1. **Develop a Digital Twin Environment:** Create a Python-based discrete-event simulation of a Flexible Job Shop that supports stochastic machine failures (modeled via Poisson distribution or Mean Time Between Failures - MTBF).
2. **Implement Benchmarking Algorithms:**
 - o *Baseline:* A Centralized **Genetic Algorithm (GA)** representing the current industry standard for static optimization.
 - o *Proposed:* A Decentralized **Multi-Agent System (MAS)** using the **Contract Net Protocol (CNP)**, where machines bid for jobs based on their real-time health and queue status.
3. **Comparative Analysis:** Quantify the "Cost of Robustness"—measuring the trade-off between the theoretical optimal speed (GA) and the realistic reliability (MAS) under varying breakdown probabilities (e.g., 5%, 10%, 20%).

4. Proposed Methodology

The research will be conducted in three distinct phases:

Phase 1: Environment Modeling (The Virtual Shop Floor)

A simulation environment will be built using Python (utilizing libraries such as SimPy or OpenAI Gym).

- **Entities:**
 - o **Jobs (Agents):** Each job carries a sequence of operations and a deadline.
 - o **Machines (Agents):** Each machine has a processing speed, a queue, and a probabilistic "Health State" (Functional, Degraded, Broken).
- **Stochasticity:** To simulate reality, random breakdown events will be injected into the system using Monte Carlo simulations.

Phase 2: Algorithm Implementation

This study compares two distinct control architectures:

Feature	Control Group (Genetic Algorithm)	Experimental Group (Multi-Agent System)
Decision Maker	One Central "Brain" (Global Solver)	Individual Machines & Jobs (Local Solvers)
Logic	Evolutionary Search	Negotiation / Bidding

	(Crossover/Mutation)	(Contract Net Protocol)
Response to Failure	Must Pause & Re-calculate entire schedule	Local Re-negotiation (Only affected node reacts)
Data Flow	Global State > Schedule	Local State > Bid > Contract

- **The Negotiation Mechanism (CNP):**
 1. **Task Announcement:** A Job Agent broadcasts a request: *"I need Milling Operation A."*
 2. **Bidding:** Machine Agents calculate a bid based on their current queue length and repair status. (e.g., *"I can start at t=10 and finish at t=15"*).
 3. **Contract Award:** The Job Agent evaluates bids and awards the contract to the machine offering the earliest completion time.

Phase 3: Experimentation & Metrics

The performance of both systems will be evaluated on standard benchmark datasets (e.g., Kacem or Brandimarte instances) under two conditions:

1. **Ideal Conditions:** 0% Breakdown probability.
2. **Disrupted Conditions:** 5% - 20% Breakdown probability.

Key Performance Indicators (KPIs):

- **Makespan (\$C_{\max}\$):** The time when the last job finishes.
- **Flow Time:** Average time a job spends in the system.
- **Stability Metric:** The variance in makespan across 100 simulation runs. (Lower variance = Higher Robustness).

5. Tools & Technologies

- **Programming Language:** Python 3.x
- **Simulation:** SimPy (Discrete Event Simulation) or Custom Class-based structure.
- **Optimization:** DEAP (for Genetic Algorithms), PyTorch (for Reinforcement Learning agents).
- **Visualization:** Matplotlib / Plotly for generating Gantt Charts and convergence graphs.

6. Expected Outcomes

We hypothesize that while the **Centralized Genetic Algorithm** will produce superior schedules in ideal conditions, the **Decentralized Multi-Agent System** will significantly outperform the baseline in high-failure environments. The decentralized approach is expected to demonstrate:

- **Reduced Idle Time:** Machines will dynamically "fill gaps" left by broken peers.
 - **Linear Scalability:** Adding more machines will not exponentially increase computation time (unlike in centralized systems).
-

Note for the Professor (Defense Strategy)

- **Relevance to Mechanical Engineering:** This project addresses **Production Planning and Control (PPC)**, a core mechanical domain. It modernizes the classic "Job Shop Scheduling" problem found in Operations Research curricula using Industry 4.0 tools.
- **Feasibility:** The project does not require physical hardware (which can break or be unavailable). It is a pure simulation study, ensuring that results can be generated and analyzed reliably within the academic timeframe.