# A Hybrid ACO-Reinforcement Learning Framework for Task Assignment

Sampath Kumar
sz4961@srmist.edu.in
Department of Computer Science Engineering
SRM Institute of Science and Technology
Chennai, India

Dr.Krishnaraj N
krishnan2@srmist.edu.in
Department of Networking And Communications
SRM Institute of Science and Technology
Chennai, India

April 12, 2025

**Abstract**

This paper presents a new approach on combining Ant Colony Optimization (ACO) with Reinforcement Learning (RL) techniques employing Q-learning and SARSA, to organize and allocate tasks in complex work environments. This approach addresses issues with order of task execution, diverse employee skills, employee availability, and fair workload distribution. ACO creates an initial task sequence considering dependencies and deadlines. Using the ACO determined task sequence, RL agents determine the initial assignments by basing their decisions depending on skill fit, workload distribution along with deadlines. This approach uses a multi objective hyperparameter tuning with Optuna library to balance high rate of task assignment with workload distribution."Refinement RL" component provides final passthrough by balancing the workload across the skills to ensure employees are not overloaded. Experiments reveal that in task assignment rate and workload distribution, the ACO-RL combination performs better than simple greedy approach. This study adds to ongoing research on scheduling and task assignment optimization problems in complex industrial setups.

# Introduction

Task assignment in industrial settings presents significant challenges due to constraints on employees, hours, skills, and efficiency. Traditional rule-based scheduling methods fail in dynamic environments with uncertainties like machine failures and changing market demands [2, 3].

### Challenges Across Sectors

This problem affects various sectors: factories need scheduling to prevent delays and costs; hospitals must meet patient needs while managing staff constraints; tech teams require flexibility for project changes; and call centers must maintain service despite varying demand.

### Impact of Remote Work

Remote work has further complicated scheduling by introducing considerations of time zones, employee availability, project sharing, skills, and attrition. Conventional "first come first assign" methodologies are inadequate for modern business dynamics [4].

### Proposed Hybrid Approach

This study proposes a framework (Fig. 1) combining Reinforcement Learning and Ant Colony Optimization. The approach consists of: (1) task sequencing via ACO to satisfy dependencies and restrictions; (2) initial employee assignment using Q-learning and SARSA with reward shaping based on skills, deadlines, workload balance, and resource efficiency; and (3) optimization through "RefinementRL" to maximize skill-level-based workload allocation and task assignment rates.

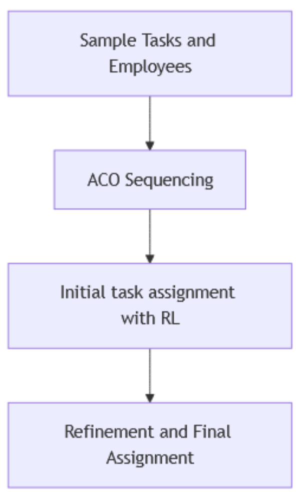


Figure 1: Proposed methodology

## Literature Survey

Reinforcement Learning (RL) has become key for workload optimization and task scheduling in dynamic environments. Ren and Liu [2] combined MachineRank with D3QN for schedule adjustment, while Zhang et al. [3] used PPO for adapting to failures. Infantes et al. [5] showed GNNs enhance DRL generalization for job-shop scheduling, and Burdett and Kozan [6] studied limited resource scheduling with heuristic models. Zhong [7] found SARSA provides stable solutions while Q-learning offers higher rewards for task assignment. Ben Noureddine et al. [8] applied multi-agent RL for distributed allocation, improving resource usage. Joo et al. [9] developed DRL considering human factors in manufacturing, Wibisono et al. [10] recommended real-time worker preference assessment, and Dastmalchian and Blyton [11] emphasized adaptable scheduling needs. Dorigo and Stützle [12] reviewed ACO for scheduling problems. Sreyas Ramesh et al. [13] demonstrated DQN's superiority for energy management. Sivamayil et al. [14] provided a cross-domain RL review, while Corazza et al. [15] compared Q-Learning and SARSA for trading systems. This study uses Optuna [17] for hyperparameter optimization and UCB strategy, which Wang et al. [18] integrated into QMTSF for cloud scheduling. Sutton and Barto [1] provide comprehensive RL algorithm guidance.

# Problem Formulation

In manufacturing and service environments, allocating tasks to a diverse workforce presents a multi-objective challenge. Our methodology integrates bio-inspired optimization, adaptive learning, and post-assignment refinement to maximize successful task assignments while ensuring deadlines are met, workloads are balanced, and skills are effectively utilized.

## Key Components

**Tasks** are characterized by:

- **Required Skills:** Specific competencies needed for completion.
- **Processing Time:** Estimated hours required.
- **Deadline:** Due date by which the task must be completed.
- **Priority:** Measure of urgency influencing scheduling order.
- **Precedence Constraints:** Dependencies between tasks.

**Employees** are defined by:

- **Skill Set:** Collection of competencies.
- **Availability:** Available work hours.
- **Efficiency:** Scaling factor influencing task completion time.
- **Current Workload:** Existing assigned tasks.

## Three-Phase Strategy

To address this NP-hard scheduling problem, our framework consists of:

1. **Initial Task Sequencing:** Employs bio-inspired optimization (ACO) to generate an initial task ordering based on urgency, processing times, and dependencies.

2. **Task Assignment through Learning:** A centralized agent dynamically assigns tasks to employees using Q-learning and SARSA algorithms.

3. **Post-Assignment Refinement:** Iteratively adjusts task-employee mapping to balance workloads and improve skill alignment.

# State and Reward Design

The reinforcement learning state is defined as:

$$s = \big(i, \mathbf{H}, \mathbf{A}, \mathbf{F}\big) \tag{1}$$

Where $i$ is the task index, $\mathbf{H}$ represents residual work hours, $\mathbf{A}$ tracks assignments, and $\mathbf{F}$ contains task features.

The reward function balances multiple objectives:

$$r(s,a) = R_{\text{feas}} + R_{\text{penalty}} + R_{\text{balance}} + R_{\text{efficiency}} \tag{2}$$

# Algorithmic Architecture

Our system integrates:

1. **Ant Colony Optimization (ACO)** for task sequencing, which generates initial sequences considering dependencies, deadlines, and priorities. The probability of transitioning from task $t_i$ to task $t_j$ is computed as:

$$p_{ij} = \frac{[\mathbf{P}_{ij}]^{\alpha}\,[\eta_{ij}]^{\beta}}{\sum_{k\in\mathcal{N}_i}[\mathbf{P}_{ik}]^{\alpha}\,[\eta_{ik}]^{\beta}} \tag{3}$$

where $\mathcal{N}_i$ is the set of candidate tasks for task $t_i$.

2. **Reinforcement Learning (RL)** for assignment, using both Q-learning:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right] \tag{4}$$

and SARSA:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma Q(s',a') - Q(s,a)\right] \tag{5}$$

with an $\varepsilon$-greedy policy.

3. **Post-Assignment Refinement** to optimize the initial assignments:

$$Q(\mathscr{S},a) \leftarrow Q(\mathscr{S},a) + \alpha_r\left[r + \gamma_r \max_{a'} Q(\mathscr{S}',a') - Q(\mathscr{S},a)\right] \tag{6}$$

The framework incorporates automated hyperparameter tuning via Optuna to optimize parameters across all stages.

# Results and Insights

Our simulation results demonstrate that:

- The combined ACO-RL approach enhances workload balance without compromising assignment rates or deadline compliance.
- Greedy methods typically overload employees with matching skill profiles, while our RL module redistributes tasks more evenly.
- The 70:30 epsilon decay schedule with Upper Confidence Bound mechanism improves reward performance by balancing exploration and exploitation.
- Though the framework improves workload distribution, practical deployment requires careful tuning to align with organizational priorities.

## Workload Distribution Insights

Results indicate that greedy assignment methods typically overload a subset of employees due to repeated allocation of tasks that match their skill profiles. By integrating a workload balancing term into the reward function, the reinforcement learning module gradually redistributes tasks more evenly, as evidenced by a lower standard deviation of assigned hours relative to baseline approaches.

## Practical Trade-offs and Future Considerations

Although the hybrid ACO-RL framework markedly improves workload distribution, its practical deployment requires careful tuning of reward parameters to align with organizational priorities. Future enhancements could include dynamic adjustment of reward components in response to real-time performance metrics.

The integration of workload balance as a primary objective contributes to more resilient and sustainable workforce management.

# Acknowledgments

The authors would like to express their sincere gratitude to SRM Institute of Technology for providing the resources and support necessary for this study.

# References


[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 1998.

[2] J. Li, L. Zhang, Z. Li, and S. Zhang, "Dynamic scheduling for flexible job shop based on MachineRank with deep reinforcement learning," *Sci. Rep.*, vol. 14, no. 1, p. 12345, 2024. doi:10.1038/s41598-024-79593-8.

[3] M. Zhang, Y. Lu, Y. Hu, N. Amaitik, and Y. Xu, "Dynamic Scheduling Method for Job-Shop Manufacturing Systems by Deep Reinforcement Learning with Proximal Policy Optimization," *Sustainability*, vol. 14, no. 9, p. 5177, 2022. doi:10.3390/su14095177.

[4] Wang B, Liu Y, Qian J, Parker SK. The remote revolution: assessing the impact of working from home on employees' productivity and work-life balance. *Future Business Journal*. 2023;9(1):1-14. doi:10.1186/s43093-024-00345-1.

[5] G. Infantes et al., "Learning to Solve Job Shop Scheduling under Uncertainty," CPAIOR, 2024.

[6] R. L. Burdett and E. Kozan, "The Assignment of Individual Renewable Resources in Scheduling," Asia Pacific Journal of Operational Research, 2004.

[7] L. Zhong, "Comparison of Q-learning and SARSA Reinforcement Learning Models on Cliff Walking Problem," DAI, 2024.

[8] D. Ben Noureddine et al., "Multi-agent Deep Reinforcement Learning for Task Allocation in Dynamic Environment," ICSOFT, 2017.

[9] T. Joo et al., "Task Allocation in Human–Machine Manufacturing Systems Using Deep Reinforcement Learning," Sustainability, 2022.

[10] A. Wibisono, A. S. Nisafani, H. Bae, and Y. Park, "A dynamic and human-centric resource allocation for managing business process execution," University of Texas at El Paso, Tech. Rep., Dec. 2016.

[11] A. Dastmalchian and P. Blyton, "Workplace flexibility and the changing nature of work: An introduction," *J. Management Studies*, vol. 38, no. 2, pp. 281-286, 2001.

[12] M. Dorigo and T. Stützle, "Ant colony optimization: Overview and recent advances," in *Handbook of Metaheuristics*, 3rd ed. Springer, 2016, pp. 311-351.

[13] S. Ramesh, S. B. N., S. J. Sathyavarapu, V. Sharma, N. K. A. A., and M. Khanna, "Comparative analysis of Q-learning, SARSA, and deep Q-network for microgrid energy management," *Scientific Reports*, vol. 15, p. 694, 2025, doi:10.1038/s41598-024-83625-8.

[14] K. Sivamayil, E. Rajasekar, B. Aljafari, S. Nikolovski, S. Vairavasundaram, and I. Vairavasundaram, "A systematic study on reinforcement learning based applications," *Energies*, vol. 16, p. 1512, 2023, doi:10.3390/en16031512.

[15] M. Corazza and A. Sangalli, "Q-Learning and SARSA: A Comparison between Two Intelligent Stochastic Control Approaches for Financial Trading," Working Paper, Ca' Foscari University of Venice, 2015. Available: `https://ssrn.com/abstract=2617630`.

[16] M. Mavrovouniotis, M. N. Anastasiadou, and D. Hadjimitsis, "Measuring the Performance of Ant Colony Optimization Algorithms for the Dynamic Traveling Salesman Problem," *Algorithms*, vol. 16, no. 545, 2023, doi: 10.3390/a16120545. Available: `https://doi.org/10.3390/a16120545`.

[17] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, *Optuna: A Next-generation Hyperparameter Optimization Framework*, Preprint, compiled July 26, 2019.

[18] Y. Wang, S. Dong, and W. Fan, *Task Scheduling Mechanism Based on Reinforcement Learning in Cloud Computing*, Mathematics, vol. 11, 3364, 2023, doi:10.3390/math11153364.

# Appendix

The source code for this project is available on GitHub at `https://github.com/samkrdev/Hybrid-ACO-RL-Task-assigner`.