# A Hybrid ACO-Reinforcement Learning Framework for Task Assignment

Sampath Kumar[1, a)] and Dr.Krishnaraj N[2, b)]

[1)]*Department of Computer Science Engineering,*
*SRM Institute of Science and Technology,*
*Chennai,*
*India*

[2)]*Department of Networking And Communications,*
*SRM Institute of Science and Technology,*
*Chennai,*
*India*

(Dated: 29 March 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.

**Keywords:** Reinforcement learning, Task scheduling, Resource allocation, Ant colony optimization, Manufacturing systems, Skill matching, Workload balancing

## INTRODUCTION

Task assignment in industrial settings, for both people or machines, is a hard problem to solve while achieving smooth operations and strong employee management. This because of the requirement of optimal task assignments taking account of the limit of employees, hours skills and efficiency. As Zhang et al. (2022) have shown, old ways of scheduling tasks according to fixed rules and simple ideas do not work in fast changing work settings, for example, heuristic based methods cannot effectively deal with dynamic job-shop environments with uncertainties such as unexpected machine failures and rapidly changing market demands [2, 3].

### Challenges Across Sectors

This problem compounds even more in different sectors that rely heavily on task planning. In factories good scheduling and assignment can stop production delays or extra costs. Hospitals need schedules that meet patient needs while matching available staff and laws. Tech teams require loose schedules to handle sudden project changes. Call centres must keep service steady while facing varying demand.

[a)]Electronic mail: sz4961@srmist.edu.in
[b)]Electronic mail: krishnan2@srmist.edu.in

## Impact of Remote Work

To make matters complicated the increase in remote and hybrid work has complicated scheduling and task assignments.Remote work offers enhanced flexibility and autonomy, allowing employees to manage their work schedules and surroundings, thereby fostering a better work-life balance [4]. Companies must take in to account: time zones, the nature of employee availability, sharing of employee between projects, their skills and frequent attrition. Old ways of scheduling and assignments based on "first come first assign" kind of greedy methodologies do not really fit well with the fast moving nature of modern businesses.

## Proposed Hybrid Approach

This study proposes a framework (Fig. 1) for the complex task assignment problem in an industrial setting with task dependencies and employee availability constraints, combining Reinforcement Learning and Ant Colony Optimization. First, we begin with task sequencing by ACO, which determines an optimal task sequence that satisfies task dependencies, due date, and priority restrictions. Then, we devise an initial employee assignment using Reinforcement Learning—specifically, Q-learning and SARSA (for comparison). Reward shaping guides these initial assignments by ensuring skills match, deadlines are met, work is balanced, and resources are used efficiently. Finally, a custom component named “RefinementRL” takes these initial assignments and optimizes them using a state representation that captures the workload, assignment, and unassigned tasks. Ultimately, the goal is to maximize a skill-level-based workload allocation and task assignment rate.

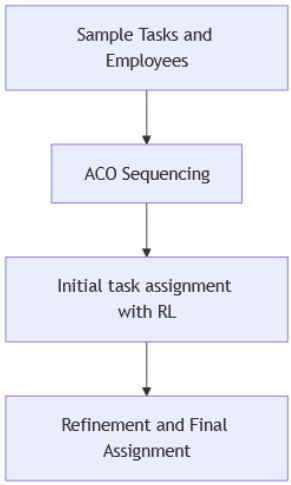


**FIGURE 1.** Proposed methodology

## Significance

Significance of this study comes from the integration of Reinforcement Learning with Ant Colony Optimization in task assignment optimization problem, which approaches the focus of critical workforce management and resource allocation issue. It demonstrates how meta heuristic algorithms can improve the current reinforcement learning approaches in complex scheduling environment.

The general applicability of the framework is largely important due to its wide use in various industries, for example manufacturing to software development where it is important for resource constrained scheduling and skilled task assignment. Additionally, the framework allows RL component to continuously learn from experience and the ACO component to manage task sequencing in an efficient manner which is a major progress compared to traditional

static methods of assignments. The practical solutions provided in this paper to industry-wide resource optimization problems contribute to both the academic literature on hybrid AI systems.

## LITERATURE SURVEY

Reinforcement Learning (RL) has developed into a promising technique for dynamic environment workload optimization and task scheduling during the past years. Traditional heuristics and rule-based methods lose their value as scheduling problems become more advanced so adaptive RL-based approaches step in to replace them. The paper examines previous studies which examine reinforcement learning alongside associated optimization approaches for workload distribution and task distribution and scheduling in uncertain environments.

Dynamic scheduling in job shops is difficult due to unexpected events. Ren and Liu [2] have developed a method that combines MachineRank assessments of machine utilization (including incoming jobs for processing at machines and outgoing ones) with D3QN to dynamically adjust schedules to improve the maximum completion times and increase on-time job completions.

In similar scenario, Zhang et al. [3] have used Proximal Policy Optimization (PPO), a form of Deep Reinforcement Learning, to focus their efforts. Their approach adapted quickly to actual failures when tested with them, whereas conventional techniques lagged behind.

The research by Infantes et al. [5] explores Deep Reinforcement Learning (DRL) solutions for Job-Shop Scheduling Problems (JSSP) under uncertain conditions. GNNs enable the enhancement of DRL model generalization and scalability according to their research. DRL proves successful for managing unpredictable task duration fluctuations and outperforms conventional deterministic methods according to their examination.

Burdett and Kozan [6] study scheduling with limited resources and renewable resource assignment. The authors applied heuristic-based mathematical models for workforce allocation to develop constraint models that may be integrated into RL-based scheduling systems.

The paper by Zhong [7] examines how Q-learning performs against SARSA reinforcement learning methods for task assignment. SARSA delivers stable solutions through its on-policy approach yet Q-learning achieves higher but riskier rewards by using its off-policy exploration method. Experts need these understandings to develop RL models which effectively manage exploration against exploitation for actual scheduling situations.

Ben Noureddine et al. [8] introduces multi agent reinforcement learning through Deep Q-learning to handle distributed task allocation. Through proposed agent teamwork operators gain better resource usage and less conflict during scheduling. The method proves beneficial for distributed systems which lack access to global information.

The research by Joo et al. [9] applies reinforcement learning methods to human-machine manufacturing systems. The research team developed a DRL model to handle human fatigue and skill abilities when performing dynamic task assignments. The research shows how human performance limitations integrated into RL-based scheduling systems produces more efficient operations and lowers worker exhaustion levels.

The research by Wibisono et al. [10] introduces a human-centered strategy for allocating resources during business process execution. The research supports RL-based scheduling by recommending real-time worker preference and availability assessment during scheduling operations.

Dastmalchian and Blyton [11] analyze how modern work patterns require adaptable scheduling systems. The research demonstrates that workforce adaptability needs to become part of all future task allocation models.

Dorigo and Stützle [12] deliver an extensive review of Ant Colony Optimization (ACO) and its scheduling problem applications. The bio-inspired optimization method ACO demonstrates great effectiveness when solving hard-to-tackle optimization problems. The review presents hybridization methods which use reinforcement learning to enhance task allocation efficiency.

Sreyas Ramesh *et al.* [13] investigate the application of Q-learning, SARSA, and deep Q-networks (DQN) for microgrid energy management. The authors compare these reinforcement learning methods in a simulated microgrid environment and demonstrate that while DQN achieves superior performance, traditional techniques like Q-learning and SARSA offer advantages in terms of computational simplicity and ease of implementation.

Sivamayil *et al.* [14] provide a systematic study of reinforcement learning applications across diverse domains, including energy management, robotics, and finance. Their review categorizes RL methodologies and highlights that the choice between value-based and policy-based approaches, such as Q-learning and SARSA, should be guided by the specific dynamics and requirements of the application at hand.

In [15], the authors address a stochastic control problem by comparing two reinforcement learning algorithms: Q-Learning and SARSA. The study focuses on developing an intelligent trading system that operates under the Adap-

tive Market Hypothesis. Both algorithms are tested on simulated and real stock price time series, with the research emphasizing the trade-offs between off-policy and on-policy learning methods in adapting to non-stationary market conditions. Their results demonstrate that while both methods are capable of adapting trading strategies based on real-time feedback, nuances in learning rates and exploration strategies critically impact overall performance.

For this study we use Optuna by T. Akiba et al [17]. Optuna is a next generation hyperparameter optimization framework which provides a define-by-run API to build complex search spaces, and easily tune hyperparameters in an interactive as well as a distributed manner, compared to the static static methods.

We also use Upper Confidence Bound (UCB) strategy. In their study [18], Wang et al propose a new reinforcement learning (RL) framework, Q-learning based Multi-Task Scheduling Framework (QMTSF), for cloud task scheduling. Their approach consists of a two-stage process where tasks are first dynamically allocated to appropriate servers and then scheduled to virtual machines (VMs) using an enhanced Q-learning algorithm that incorporates the Upper Confidence Bound (UCB) strategy.By leveraging the UCB strategy—which balances exploration (trying new actions) and exploitation (using known rewards) and is more efficient than traditional methods such as the epsilon-greedy approach—the algorithm achieves faster convergence. Consequently, QMTSF reduces the overall makespan (i.e., the total time to complete all tasks) and average task processing time, as well as enhances energy efficiency and CPU utilization in the dynamic and heterogeneous cloud computing environment.

Sutton and Barto's *Reinforcement Learning: An Introduction* [1] is a go-to guide for anyone curious about Reinforcement Learning. This scholarly work elaborates algorithms like Q learning and policy iteration in detailed manner.

## PROBLEM FORMULATION

In modern manufacturing and service environments, the allocation of tasks to a diverse workforce poses a complex, multi-objective challenge. The aim is to maximize the number of tasks successfully assigned while ensuring that deadlines are met, workloads are balanced, and employees' skills are effectively utilized. To address this, our methodology decomposes the problem into distinct but interrelated phases, integrating bio-inspired optimization, adaptive learning, and post-assignment refinement.

### Task Characteristics

Each work item is characterized by several intrinsic properties:

- **Required Skills:** The specific competencies needed to complete the task.
- **Processing Time:** An estimate of the hours required for completion.
- **Deadline:** The due date by which the task must be finalized.
- **Priority:** A measure of urgency that influences scheduling order.
- **Precedence Constraints:** Dependencies indicating that certain tasks must be completed before others can begin.

These attributes introduce both temporal and logical constraints that must be considered in the scheduling process.

### Employee Attributes

The available workforce is described by:

- **Skill Set:** The collection of competencies held by each employee.
- **Availability:** The number of work hours an employee can commit within a given period.
- **Efficiency:** A scaling factor that influences the effective time required to complete tasks.

- **Current Workload:** An evolving measure of the tasks already assigned to the employee.

A candidate employee is deemed suitable for a task if there is an overlap between the task's required skills and the employee's capabilities, and if the employee has sufficient available hours to accommodate the task, accounting for individual efficiency.

## Hybrid Scheduling and Assignment Strategy

To navigate the NP-hard nature of this scheduling problem, our framework is structured into three sequential phases:

1. **Initial Task Sequencing via Bio-inspired Optimization:** A pheromone-based search process is employed to generate an initial ordering of tasks. This stage incorporates heuristic evaluations that combine task urgency, estimated processing times, deadlines, and dependency considerations. The generated sequence is periodically updated to reflect improvements in assignment performance as subsequent stages progress.

2. **Task Assignment through Adaptive Learning:** Building upon the sequenced tasks, a centralized learning agent dynamically assigns tasks to employees. This process considers the current state of assignments, available work hours, and skill matches, and it is designed to adapt through an exploration–exploitation mechanism. Q -learning and SARSA are applied to refine decision-making during the assignment process. We are using these two algorithms to compare them.

3. **Post-Assignment Refinement:** Following the initial allocation, a dedicated refinement module iteratively adjusts the task-to-employee mapping. The objective of this stage is to further balance workloads and improve the alignment of tasks with employee skills, ensuring that operational constraints are met and overall efficiency is maximized.

## Automated Hyperparameter Tuning

An automated tuning process is integrated into the framework to optimize key parameters across all stages. Using a multi-objective optimization approach, the tuner adjusts factors influencing the bio-inspired sequencing, the learning dynamics in the assignment phase, and the iterative refinement process. The objective is to enhance task coverage while simultaneously reducing variability in performance metrics, thus ensuring a robust and adaptive solution.

# STATE AND REWARD DESIGN

## Reinforcement Learning Framework

Let $N$ be the total number of employees. We define the RL state as

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

where

- $i$: index in the task sequence,
- $\mathbf{H} = \big[h_1, \ldots, h_N\big]$: residual work hours per employee,
- $\mathbf{A}$: binary record of which tasks have been assigned,
- $\mathbf{F}$: binned features such as priority or due date.

At each step, the agent receives a reward:

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

$$R_{\text{feas}}(s,a) = \lambda\, p_t\, \mathbb{I}\{\text{feasible}\} \tag{3}$$
$$R_{\text{penalty}}(s,a) = -\kappa\, \mathbb{I}\{\text{violation}\} - \nu\, \mathbb{I}\{\text{priority skip}\} \tag{4}$$
$$R_{\text{balance}}(s,a) = \alpha\Big(1 - \tfrac{\sigma(W)}{\overline{W}}\Big) \tag{5}$$
$$R_{\text{efficiency}}(s,a) = \eta\, \tfrac{\Delta\tau}{\tau} \tag{6}$$

where $\lambda, \kappa, \nu, \alpha, \eta$ are weighting constants; $p_t$ indicates the task's priority; $\sigma(W), \overline{W}$ are the standard deviation and mean of assigned hours; and $\Delta\tau/\tau$ measures how much an employee's efficiency reduces task duration.

## Refinement Phase

A subsequent refinement step uses an augmented state and reward:

$$S = \big(\mathbf{W}, M, U\big) \tag{7}$$

where $\mathbf{W} = [w_1, \ldots, w_N]$ captures each employee's total workload, $M$ is the mapping of tasks to employees, and $U$ is the set of unassigned tasks.

The refinement reward is:

$$r_{\text{ref}}(S,a) = \beta_1 R_{\text{match}} - \beta_2 R_{\text{imbalance}} + \beta_3 R_{\text{eff}} \tag{8}$$

where $\beta_1, \beta_2, \beta_3$ weight the emphasis on skill–task alignment ($R_{\text{match}}$), workload fairness ($R_{\text{imbalance}}$), and additional efficiency gains ($R_{\text{eff}}$).

**TABLE I.** Notation and Descriptions

| Symbol | Description |
|---|---|
| $i$ | Index in the task sequence |
| $\mathbf{H}$ | Residual hours vector, $[h_1, \ldots, h_N]$ |
| $\mathbf{A}$ | Binary assignment record |
| $\mathbf{F}$ | Binned task features (priority, due date) |
| $s$ | RL state in (1) |
| $r(s,a)$ | RL reward in (2) |
| $\lambda, \kappa, \nu, \alpha, \eta$ | RL weighting constants |
| $p_t$ | Priority of the current task |
| $\sigma(W), \overline{W}$ | Std. dev. and mean of assigned hours |
| $\frac{\Delta\tau}{\tau}$ | Relative reduction in task duration (efficiency) |
| $S$ | Refinement state in (7) |
| $\mathbf{W}$ | Workloads: $[w_1, \ldots, w_N]$ |
| $M$ | Task–employee mapping |
| $U$ | Set of unassigned tasks |
| $r_{\text{ref}}(S,a)$ | Refinement reward in (8) |
| $\beta_1, \beta_2, \beta_3$ | Refinement reward coefficients |

# ALGORITHMIC ARCHITECTURE AND DESIGN

In dynamic operational environments, the challenge of scheduling tasks and allocating them to a diverse workforce demands a robust and adaptable framework. Our system integrates heuristic optimization via Ant Colony Optimization

(ACO), centralized reinforcement learning (RL) for task assignment, and an iterative post-assignment refinement process—all underpinned by automated hyperparameter tuning using Optuna. The architecture diagram (Fig. 1) provides a detailed logic and data flow.

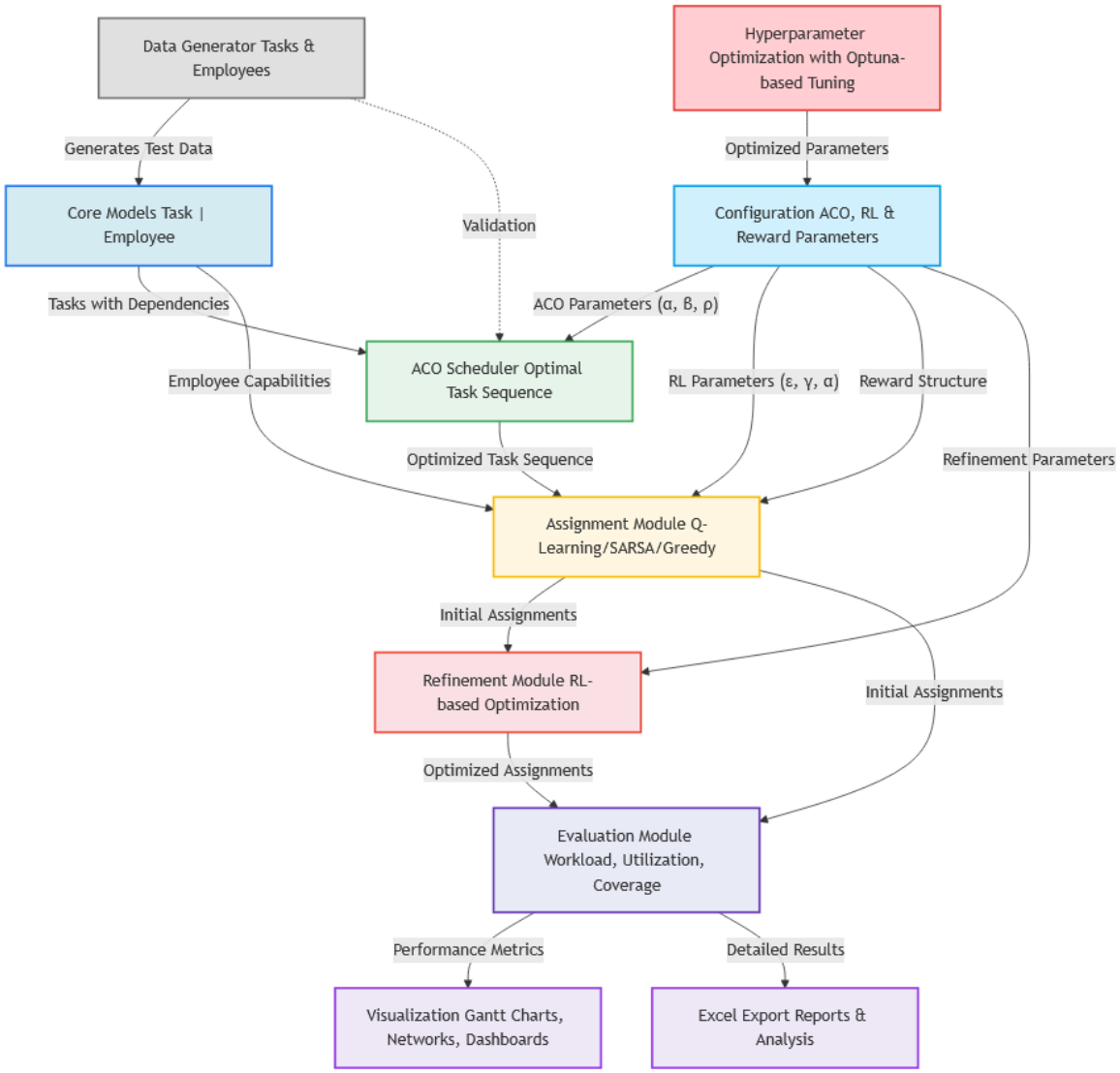


**FIGURE 2.** Architecture Diagram

## Framework Overview

The architecture is organized into three core modules:

1. **Task Sequencing:** An ACO-based module generates an initial task sequence by considering inter-task dependencies, deadlines, and priorities. The module leverages configurable parameters (e.g., pheromone influence $\alpha$,

heuristic weight $\beta$, evaporation rate $\rho$, etc.) to guide the search.

2. **RL-based Assignment:** Centralized RL agents (employing both Q-learning and SARSA variants) assign tasks to employees. The state representation includes the current task index, discretized measures of employee workload, and historical assignment data. An $\varepsilon$-greedy policy—with optional Upper Confidence Bound (UCB) enhancements—is used to balance exploration and exploitation.

3. **Post-Assignment Refinement:** A dedicated refinement module further optimizes the initial assignment. By iteratively adjusting the task-to-employee mapping based on updated workload distributions and unassigned tasks, this stage improves overall balance and skill alignment.

## Task Sequencing via Ant Colony Optimization (ACO)

The ACO module starts by initializing a pheromone matrix $\mathbf{P}$ with parameters such as $\alpha$, $\beta$, and $\rho$ (all configurable via the system settings). Each task is assigned a heuristic value $\eta_{ij}$ based on factors like priority, estimated duration, and deadline constraints. The probability of transitioning from task $t_i$ to task $t_j$ is computed as:

$$p_{ij} = \frac{[\mathbf{P}_{ij}]^{\alpha}\,[\eta_{ij}]^{\beta}}{\sum_{k\in\mathcal{N}_i}[\mathbf{P}_{ik}]^{\alpha}\,[\eta_{ik}]^{\beta}} \tag{9}$$

where $\mathcal{N}_i$ is the set of candidate tasks for task $t_i$. After constructing candidate sequences with multiple agents, the pheromone levels are updated by:

$$\mathbf{P}_{ij} \leftarrow (1-\rho)\,\mathbf{P}_{ij} + \Delta\mathbf{P}_{ij} \tag{10}$$

with $\Delta\mathbf{P}_{ij}$ reflecting the quality of the sequence. This process is re-invoked periodically (e.g., every few hundred training episodes) to adapt to evolving system conditions.

## RL-based Task Assignment and Post-Assignment Refinement

The RL module formulates task assignment as a sequential decision process. The state $s$ captures the current task index, residual working hours for each employee, and a history of past assignments. Actions include assigning a task to an employee or opting to skip it if constraints cannot be met. For Q-learning, the update rule is:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right] \tag{11}$$

and for SARSA:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma Q(s',a') - Q(s,a)\right] \tag{12}$$

A scheduled decay of the exploration rate $\varepsilon$, along with optional UCB adjustments, guides the learning process. Following the initial assignment, the *RefinementRL* module further optimizes the allocation by considering an updated state $\mathscr{S}$ (comprising current workloads, the mapping of tasks to employees, and unassigned tasks) and applying a similar RL update:

$$Q(\mathscr{S},a) \leftarrow Q(\mathscr{S},a) + \alpha_r\left[r + \gamma_r \max_{a'} Q(\mathscr{S}',a') - Q(\mathscr{S},a)\right] \tag{13}$$

Here, the refinement parameters $\alpha_r$ and $\gamma_r$ are tuned specifically to improve workload balance and alignment with employee skills.

**TABLE II.** Legend of Symbols

| Symbol | Description |
|---|---|
| $i$ | Current task index |
| $\mathbf{P}$ | Pheromone matrix |
| $\eta_{ij}$ | Heuristic value for transitioning from task $t_i$ to $t_j$ |
| $\rho$ | Pheromone evaporation rate |
| $s, s'$ | RL states before and after an action |
| $a, a'$ | Actions chosen in the current and subsequent states |
| $\alpha, \alpha_r$ | Learning rates for initial assignment and refinement |
| $\gamma, \gamma_r$ | Discount factors for initial assignment and refinement |
| $\varepsilon$ | Exploration rate in RL |
| $\mathbf{H}$ | Vector of residual working hours |
| $r$ | Immediate reward signal |
| $\mathscr{S}$ | State representation for refinement |
| $M$ | Task-to-employee mapping |
| $U$ | Set of unassigned tasks |

## Hyperparameter Optimization and Experimental Evaluation

A standout feature of the design is the integration of automated hyperparameter tuning using Optuna. This module optimizes key parameters for both the ACO and RL components (such as learning rates, discount factors, exploration decay, and pheromone parameters) through a multi-objective process. The objectives are to maximize the proportion of tasks assigned and to minimize the variance of the reward signal. Extensive experimental trials yield Pareto-optimal configurations, with detailed visualizations and reports generated for comprehensive performance analysis.

# INFERENCE AND DISCUSSION

Based on the simulation results we can notice that generated tasks are sorted by ACO step (fig. 3) and also it combined with RL steps enhances the balance of task assignments among employees without compromising overall assignment rates or deadline compliance (figs. 4 to 7).

## Workload Distribution Insights

Results from the simulation (Fig. 4) indicate that greedy assignment methods typically overload a subset of employees due to repeated allocation of tasks that match their skill profiles. This concentration not only creates bottlenecks but also elevates the risk of fatigue and reduces overall efficiency. By integrating a workload balancing term into the reward function, the reinforcement learning module gradually redistributes tasks more evenly. As a result, the refined policies exhibit a substantial reduction in workload variance, as evidenced by a lower standard deviation of assigned hours relative to baseline approaches. This indicates that ‘RefinementRL‘ is working as intended. Notice how the Boring and Drilling tasks are handled in Greedy vs Refined Q-learning or SARSA (figs. 5 to 7). Refined algorithms display better work balancing.

## Reward Dynamics

During both the initial and refinement stages, the reward structure used by the reinforcement learning components directly improved the workload balance. The reward functions penalized excessive task concentration instead of just encouraging rapid task completion tied to strict deadlines. Through multiple training episodes, the RL agents learned and internalized the cost of overburdening certain employees, which led to more even distribution of work and achieved the dual goals of efficiency and fairness in the scheduling process.

The 70:30 epsilon decay schedule and the UCB mechanism complemented each other to improve reward performance. The UCB mechanism gave an exploration bonus to actions that were less visited, so that the agent would explore other actions before declaring an action optimal. The agents were able to learn better action value estimates earlier in this balanced exploration. In the meantime, the 70:30 decay schedule decreased the exploration rate during the first 70% of the episodes (exploration vs exploitation) and kept a stable low exploration regime in the last 30% of training. Together these strategies let the agents learn more robustly, with a steadily increasing moving average of rewards during training(fig. 8).

## 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. In certain scenarios, urgent tasks might necessitate a temporary deviation from perfect balance in order to assign critical tasks to the most qualified personnel. Future enhancements could include dynamic adjustment of reward components in response to real-time performance metrics. Overall, the incorporation of workload balance as a primary objective not only advances scheduling methodologies but also contributes to more resilient and sustainable workforce management.

## Plots from the Simulation

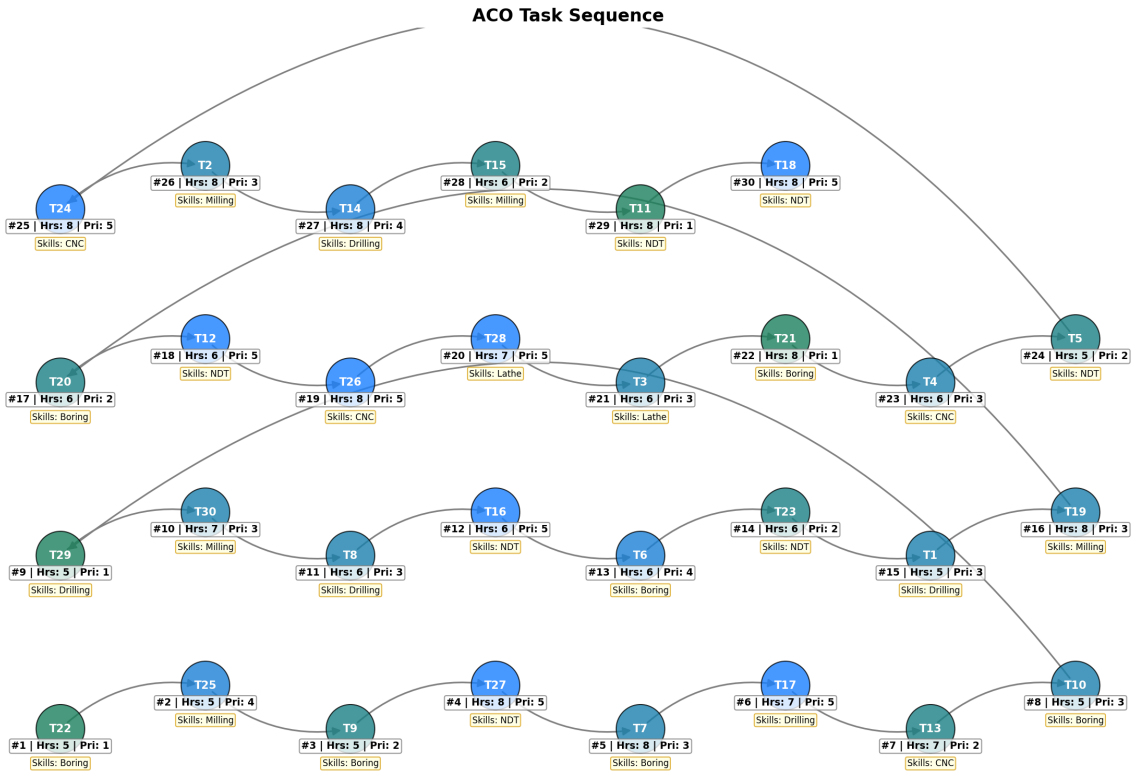


**FIGURE 3.** Illustration of the Ant Colony Optimization (ACO) task sequence.

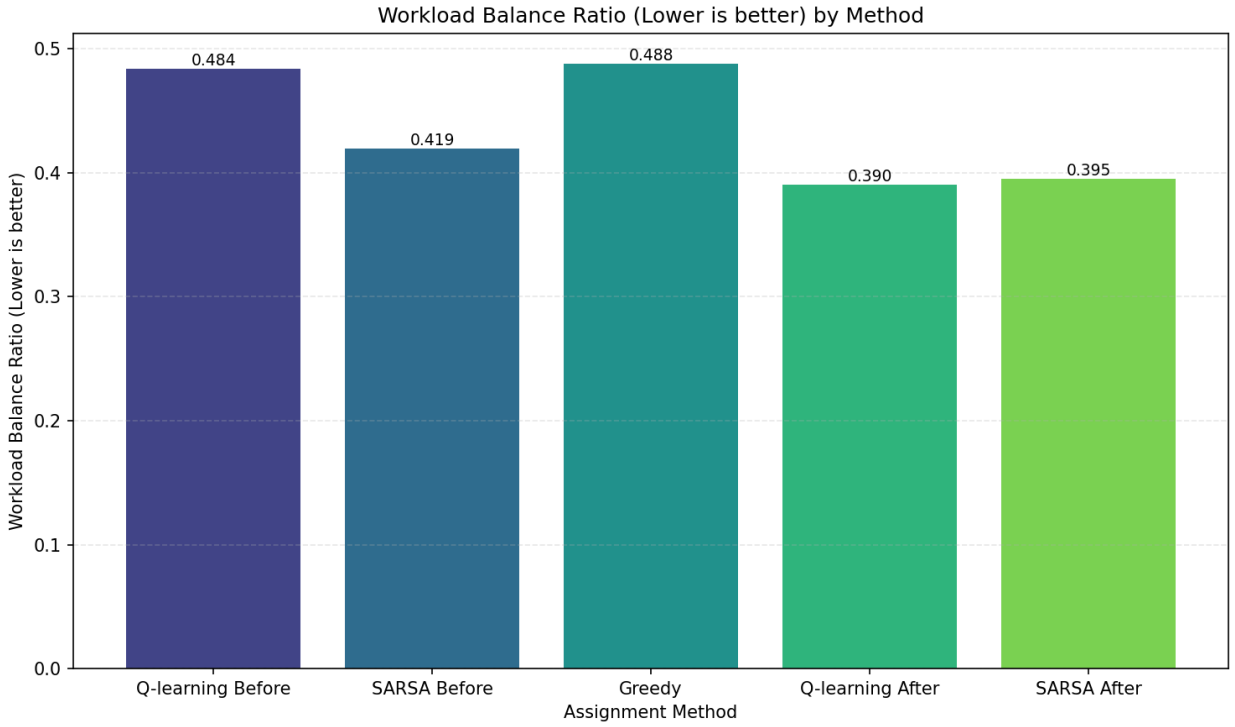


**FIGURE 4.** Workload Balance Ratio (lower is better) by assignment method.

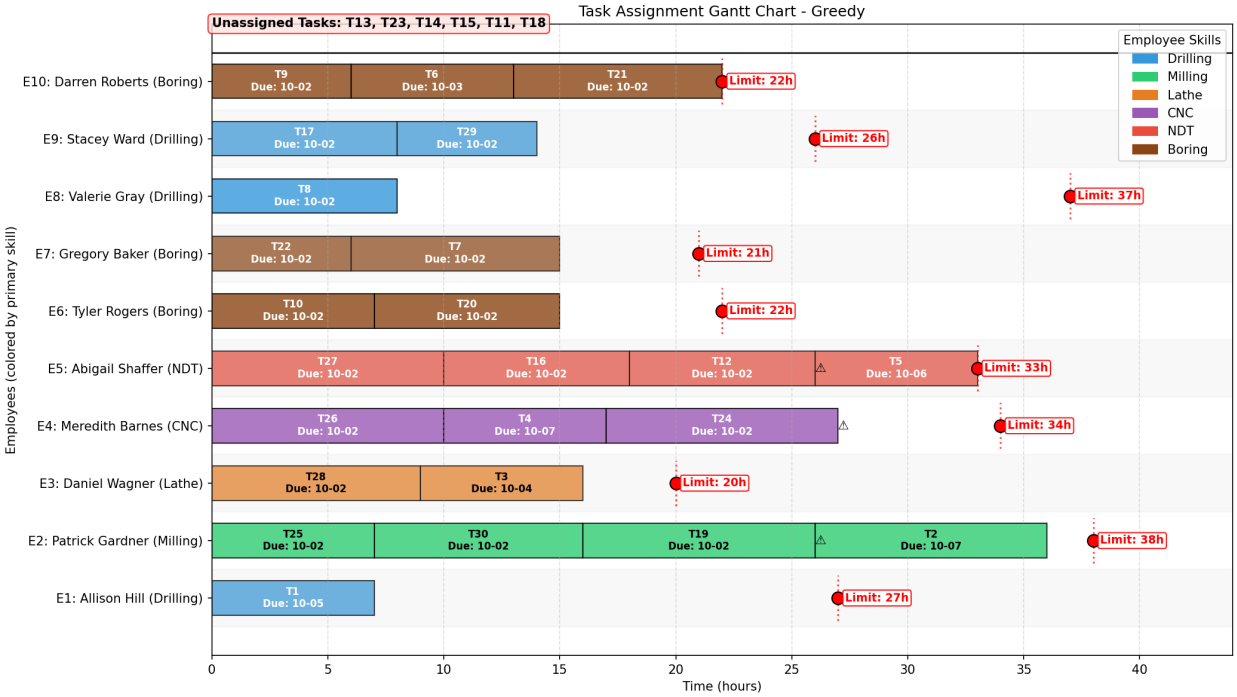


**FIGURE 5.** Assignments by Greedy method

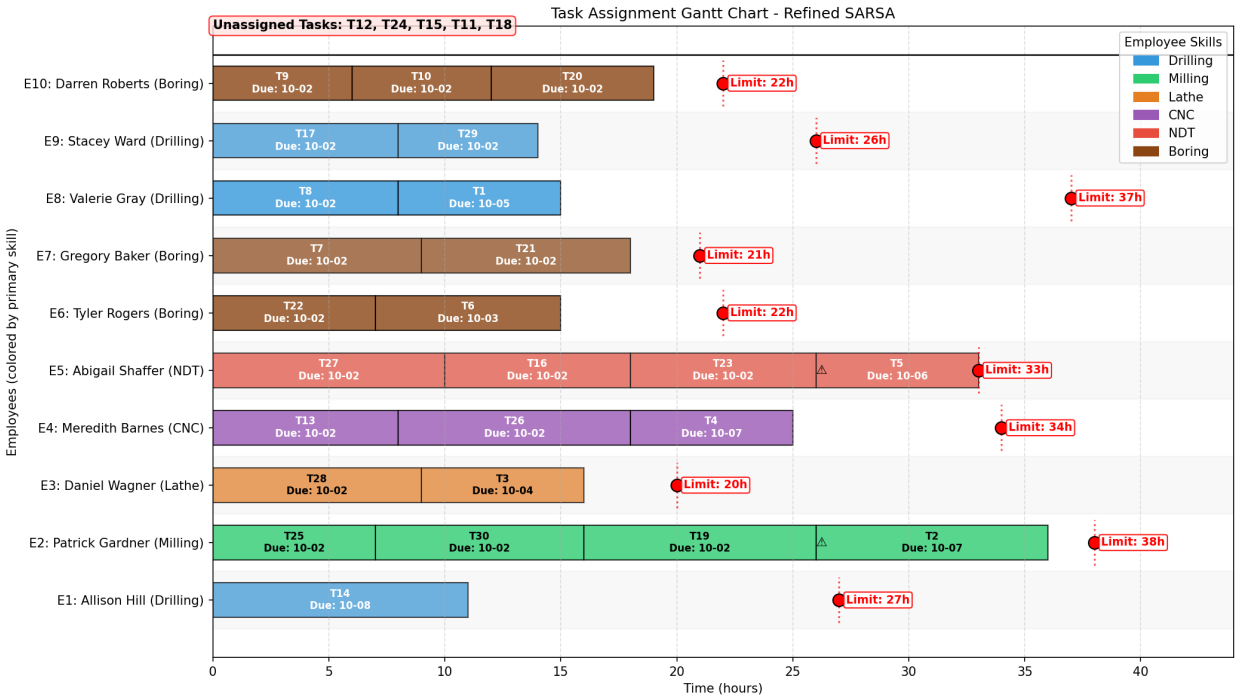


**FIGURE 6.** Assignments by SARSA after Refinement

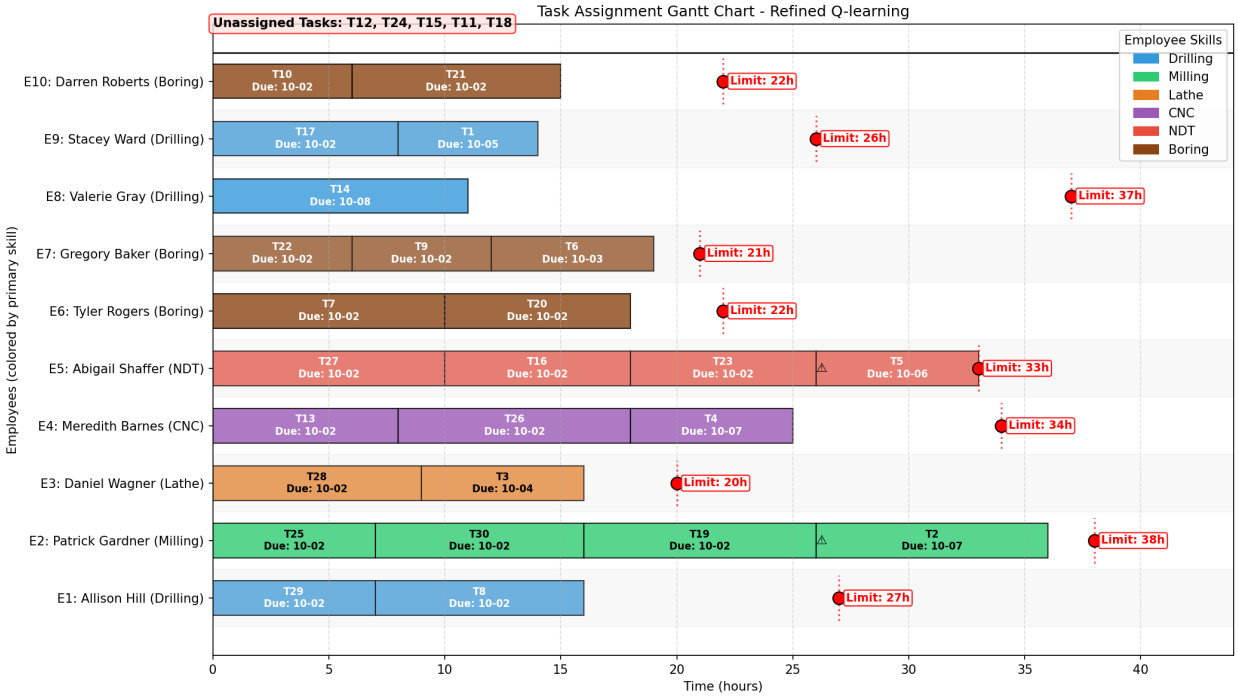


**FIGURE 7.** Assignments by Q-learning after Refinement

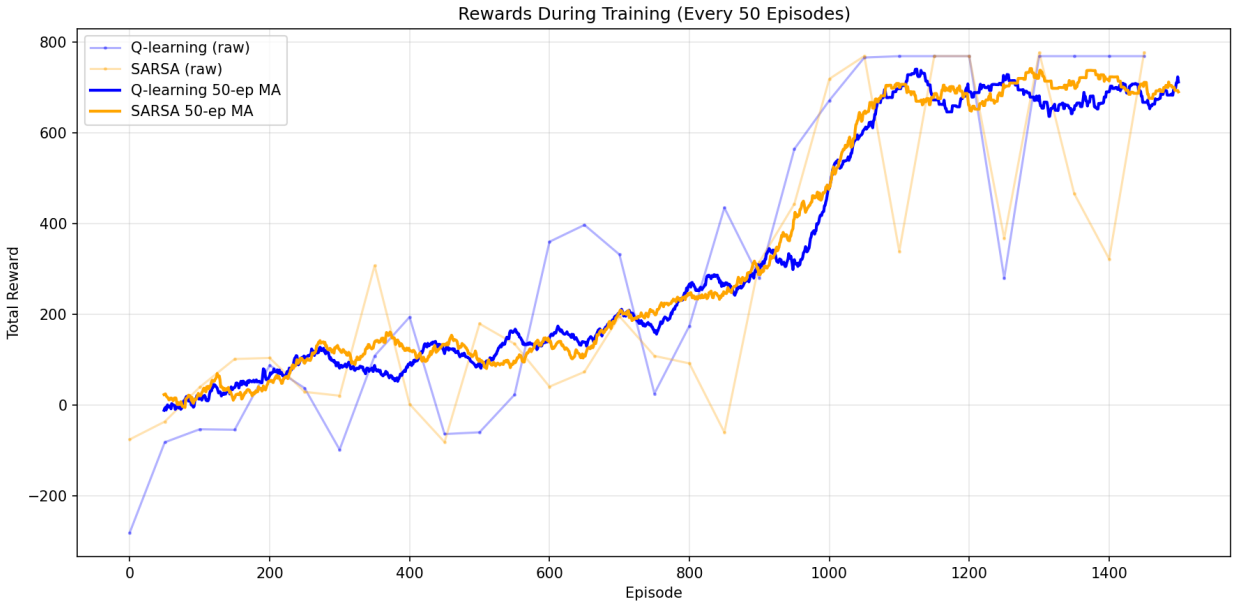


**FIGURE 8.** Rewards during training (with 50 MA).

## FUTURE WORK

While the current framework—which combines ACO-based task sequencing with reinforcement learning for employee assignment—has yielded promising results, several avenues for further research remain:

- **Enhanced State Representations:** The present RL models utilize discretized state representations. Future research could explore richer representations, such as deep neural network embeddings, to capture complex task dependencies and nuanced employee characteristics more effectively.

- **Hybrid Optimization Techniques:** Integrating additional meta-heuristic or evolutionary algorithms with ACO may lead to improved task sequencing. Investigating hybrid models that merge global search capabilities with localized refinement strategies could further optimize scheduling performance.

- **Adaptive Reward Mechanisms:** Given the critical role of the reward structure, future work could develop dynamic or adaptive reward schemes that adjust penalties and incentives based on real-time feedback or historical data, thereby enhancing both workload balance and resource utilization.

- **Scalability and Real-Time Implementation:** Although our experiments were conducted on a moderate scale, real-world applications often involve larger task sets and employee pools. Future studies should address scalability challenges and explore real-time implementations, possibly leveraging distributed computing frameworks.

- **Incorporating Uncertainty:** Extending the framework to account for uncertainties in task durations, employee availability, and environmental changes would increase its robustness. Incorporating probabilistic models or robust optimization techniques into the RL component may better handle stochastic variations.

- **User Feedback Integration:** Future systems could incorporate continuous feedback from domain experts and end-users, enabling a human-in-the-loop approach that iteratively refines both task sequencing and assignment processes based on practical, real-world insights.

These research directions aim to bridge the gap between theoretical scheduling models and practical, scalable solutions for dynamic, multi-agent systems. By addressing these challenges, future work can further enhance the adaptability and effectiveness of the hybrid framework in diverse industrial environments.

## 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. We are especially grateful to Professor Dr. Krishnaraj N for his invaluable guidance, insightful suggestions, and continuous encouragement throughout the course of this study. His expertise and constructive feedback have been instrumental in shaping the direction and success of this work.

## 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`.