DOI 10.51885/1561-4212_2025_3_182
IRSTI 20.15.13

COMPARATIVE ANALYSIS OF OPTIMIZATION MODELS FOR RESOURCE ALLOCATION IN UNCERTAIN IT ENVIRONMENTS

АНЫҚТАЛМАҒАН IT ОРТАЛАРЫНДА РЕСУРСТАРДЫ БӨЛУДІҢ ОҢТАЙЛАНДЫРУ МОДЕЛЬДЕРІНЕ САЛЫСТЫРМАЛЫ ТАЛДАУ

СРАВНИТЕЛЬНЫЙ АНАЛИЗ МОДЕЛЕЙ ОПТИМИЗАЦИИ ДЛЯ РАСПРЕДЕЛЕНИЯ РЕСУРСОВ В УСЛОВИЯХ НЕОПРЕДЕЛЁННОСТИ IT-СРЕДЫ

N.S. Yesmukhamedov[1], S.Z. Sapakova[1], B.K. Sinchev[1], N.S. Yesmukhamedov[1]
[1]International Information Technology University, Almaty, Kazakhstan
* Corresponding author: Saya Sapakova, e-mail: s.sapakova@iitu.edu.kz

Keywords:
optimization, resource allocation, dynamic problems, mathematical models, integer programming, multi-agent systems, Markov processes.

ABSTRACT
This study addresses the challenge of resource allocation in dynamic and uncertain environments by evaluating five mathematical models: Markov Decision Process (MDP), Integer Programming, Multi-Agent Systems, Scheduling Problems, and Assignment Models. The relevance of this problem lies in the need for effective decision-making tools under time and resource constraints, particularly in IT and service-oriented industries where operations are often unpredictable and require adaptive optimization strategies. The main aim of this research is to perform a comparative analysis of the selected models using performance metrics such as execution time, cost, and memory consumption. The research methodology involves conducting a series of simulations on synthetically generated datasets that reflect realistic operating conditions in terms of workload variability, task dependencies, and limited resources. Results indicate that Integer Programming delivers the lowest solution cost but requires significantly more execution time, making it suitable for scenarios where accuracy outweighs speed. On the other hand, MDP and Multi-Agent models offer faster computation and flexibility, but with relatively higher solution costs. Scheduling and Assignment Models demonstrate a balanced trade-off but are less scalable under complex constraints. These findings highlight the inherent trade-off between time efficiency, computational complexity, and solution optimality. The proposed comparison contributes to identifying suitable approaches based on task-specific priorities and operational goals. The theoretical significance of the work lies in the integration of multiple optimization techniques into a structured comparative framework, enhancing the understanding of model behaviour under uncertainty. The practical contribution focuses on guiding system architects and decision-makers in selecting the most appropriate modelling tools for real-time systems in IT and service environments. Future work will explore enhancements to scalability, hybrid modelling strategies, and adaptive algorithms to support industrial-scale implementations with dynamic data inputs.

Түйінді сөздер:
оңтайландыру, ресурстарды бөлу, динамикалық есептер, математикалық модельдер, бүтін сандармен программалау, көп агентті жүйелер, Марков процесстері.

ТҮЙІНДЕМЕ

Бұл зерттеу жұмысы динамикалық және белгісіз ортада ресурстарды бөлудің тиімді жолдарын қарастырып, бес түрлі математикалық модельге салыстырмалы талдау жүргізуді мақсат етеді. Қарастырылған модельдер: Марков шешім қабылдау процесі (MDP), бүтінсандық бағдарламалау, көпагентті жүйелер, жоспарлау есептері және тағайындау модельдері. Зерттеліп отырған мәселенің өзектілігі – IT және қызмет көрсету салаларында уақыт пен ресурстар шектеулі жағдайда тиімді шешім қабылдау құралдарының қажеттілігінде. Бұл салада процестер жиі өзгеріп отырады, сондықтан оңтайландыруға бейімделген тәсілдер қажет. Зерттеудің негізгі мақсаты – көрсетілген модельдерді орындау уақыты, шешім құны және жад пайдалану сияқты критерийлер бойынша салыстыру. Зерттеу әдістемесі – нақты жағдайларды ескере отырып жасалған синтетикалық деректер негізінде бірқатар модельдеу тәжірибелерін жүргізуге негізделген. Нәтижелер бүтінсандық бағдарламалау моделінің шешімнің ең төменгі құнын қамтамасыз ететінін көрсетті, бірақ оны орындау уақыты басқа әдістерге қарағанда көбірек. MDP және көпагентті жүйелердің орындалу жылдамдығы жоғары, бірақ шешім құны қымбатырақ. Жоспарлау және тағайындау модельдері нәтижелері орташа, алайда күрделі жағдайларда масштабталуы қиындау. Зерттеудің теориялық маңыздылығы – бірнеше оңтайландыру әдістерін салыстырмалы құрылымға біріктіру. Ал практикалық маңыздылығы – нақты уақыт режимінде жұмыс істейтін жүйелер үшін модельдерді таңдауға әдістемелік негіз беруінде. Болашақта зерттеу бағыттары модельдердің икемділігін және өндірістік ауқымдағы бейімделу мүмкіндіктерін арттыруға бағытталады.

Ключевые слова:
оптимизация, распределение ресурсов, динамические задачи, математические модели, целочисленное программирование, мультиагентные системы, марковские процессы.

АННОТАЦИЯ

Данное исследование посвящено решению задачи распределения ресурсов в условиях динамичной и неопределённой среды путём сравнительного анализа пяти математических моделей: марковский процесс принятия решений (MDP), целочисленное программирование, многоагентные системы, задачи расписания и модели назначения. Актуальность проблемы заключается в необходимости эффективных инструментов поддержки принятия решений в условиях ограниченности времени и ресурсов, особенно в IT-сфере и сервисных отраслях, где процессы часто носят нестабильный характер и требуют адаптивных подходов к оптимизации. Цель исследования — провести сопоставительный анализ указанных моделей по таким критериям, как время выполнения, стоимость решения и потребление памяти.

Методология исследования включает серию симуляционных экспериментов на синтетически сгенерированных данных, отражающих реальные условия функционирования: изменчивую нагрузку, зависимость задач и ограниченные ресурсы. Результаты показали, что модель целочисленного программирования обеспечивает наименьшую стоимость решения, но требует больше времени на выполнение, что делает её применимой в сценариях, где приоритетом является точность. Модели на основе MDP и многоагентного подхода демонстрируют более быстрое выполнение, но с повышенными затратами. Модели расписания и назначения показывают сбалансированные результаты, но хуже масштабируются при усложнении

условий. Выводы исследования подчёркивают необходимость выбора модели на основе приоритетов – скорости или минимизации затрат. Теоретическая значимость работы заключается в объединении различных методов оптимизации в единую сравнительную структуру. Практическая ценность состоит в предоставлении рекомендаций по выбору подходящих инструментов моделирования для систем реального времени. В дальнейшем планируется расширение масштабируемости моделей и интеграция адаптивных алгоритмов для промышленного применения.

INTRODUCTION

Modern IT services operate in conditions of high variability: tasks enter the system at random times, performers have different availability and qualifications, and requirements for deadlines and priorities can quickly change. Such features are typical for technical support, cloud infrastructure maintenance, DevOps process support and other areas of IT services. In these conditions, it becomes critically important to effectively distribute tasks between available performers, minimizing response time, reducing costs and adhering to service level agreements (SLAs).

Traditional methods of work allocation, based on static information and predetermined schedules, are ineffective in the presence of incomplete or constantly changing information. This creates a need for new mathematical models and optimization methods that can take into account the dynamic nature of the environment, uncertainty, and the stochastic nature of incoming tasks.

The objective of this study is to develop and analyze mathematical models of task distribution in an IT environment with dynamic parameters, as well as to compare various optimization approaches: from classical heuristics and linear programming methods to stochastic models such as Markov decision processes (MDP) and modern reinforcement learning algorithms. The proposed models are focused on practical application in real-time systems and are aimed at improving the efficiency and sustainability of IT process management.

The problem of efficient task allocation among agents in dynamic and distributed computing environments remains one of the central challenges in the field of optimization and intelligent management systems. Contemporary approaches to this issue encompass a wide range of methods — from classical heuristics to reinforcement learning algorithms applied under conditions of uncertainty and stochastic variability.

LITERATURE REVIEW

The literature pays particular attention to aspects such as cooperative agent interaction, limited resource availability, time constraints, and the necessity for real-time adaptive strategies. This section provides a critical analysis of current scientific publications focusing on mathematical models and optimization methods used to solve task allocation problems across various architectures — from edge and cloud systems to clusters designed for machine learning. The issue of dynamic and cooperative task allocation in distributed environments has been actively studied in recent years in the context of enhancing the efficiency of distributed computing systems. The DC-MATA (Dynamic and Cooperative Multi-Agent Task Allocation) problem is considered within the framework of cooperative organization involving individually rational agents. The author proposes a learning-based approach that improves Nash equilibrium convergence in dynamically changing conditions (Costa, 2024).

The problem of dynamic task allocation and service migration in Edge-Cloud IoT systems is discussed in (Dynamic Task Allocation and Service Migration in Edge-Cloud IoT System Based on Deep Reinforcement Learning, 2022), where the goal is to minimize cloud load while considering migration and latency constraints. The proposed solution is based on the Deep

Deterministic Policy Gradient (DDPG) algorithm, which effectively learns in discrete action spaces. A further development of optimal task offloading strategies in edge environments is presented in (Pradhan, Tripathy & Matam, 2024), where the focus is on predicting resource availability before making offloading decisions. The reinforcement learning approach improves resource utilization and reduces idling. The optimization of task offloading in MEC environments with stochastic task arrivals is addressed in (Dynamic Task Offloading Optimization in Mobile Edge Computing Systems with Time-Varying Workloads Using Improved Particle Swarm Optimization, 2024). The authors apply a modified Particle Swarm Optimization (PSO) algorithm for adaptive task redistribution. The method outperforms Genetic Algorithms (GA) and Simulated Annealing in terms of performance. Han, Li, and Zhou (2023) propose a service migration strategy designed for edge environments with limited resources. A multidimensional Markov decision process combined with a Deep Q-Network (DQN) algorithm enables near-optimal decision-making, leading to reduced latency and energy consumption. Another important direction in the literature is multi-resource scheduling, particularly in machine learning clusters. Studies (Joe-Wong et al., 2012; Peng et al., 2018; Gu et al., 2019) emphasize that most existing systems focus on a single resource type (e.g., GPU), which compromises scheduling accuracy and efficiency in environments with resource competition. Recent works (Li et al., 2023; Noghabi et al., 2020) highlight the increasing demand for systems that enable adaptive task allocation, considering multiple resource types and the stochastic nature of task arrivals. Reinforcement learning (RL) has emerged as a critical tool for addressing these challenges, particularly in online combinatorial environments with complex constraints (Yu et al., 2021; Mao et al., 2016; Xu et al., 2023). Allocating resources to process tasks online is a complex problem, especially when resource availability is uncertain. One effective approach to address this challenge is to adapt task allocation problem formulations to account for dynamic resource availabilities. This can be achieved by using a prediction model that estimates task processing times for all authorized resources. The proposed adaptations have been evaluated and compared with traditional allocation strategies, such as shortest queue, random, and round-robin methods, showing better performance (Kunkler & Rinderle-Ma, 2024). In scenarios where resource demand is volatile and stochastic, the uncertainty in resource availability must be considered for optimal allocation. To tackle this, a stochastic programming framework is introduced for resource distribution problems. This framework ensures that the allocation is both efficient and adaptive to changing conditions. Experimental results demonstrate that the stochastic method outperforms deterministic approaches, achieving an average improvement of 1% in expected revenue (Lee et al., 2024).

MATERIALS AND METHODS OF RESEARCH

This paper discusses various mathematical models for optimizing task allocation in dynamic IT environments. These models allow one to effectively consider uncertainty, variability, and constraints in real time. Stochastic models, integer programming problems, multi-agent systems, and queueing theories are proposed as the main optimization tools. Each of the models will be considered from the point of view of its application in real IT environments, such as technical support, CI/CD processes, and infrastructure management.

MDP (Markov Decision Process)

We consider a system in which the state of the environment at time step t is designated as s_t, action (task distribution) — a_t, and the objective function is the maximum expected reward (for example, minimizing the total downtime or SLA violations):

$$\pi^* = \arg\max_{\pi} \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)], \quad (1)$$

where:

π- strategy (policy),

$R(s_t, a_t)$- reward (e.g. penalty for SLA violation),

$\gamma \in [0,1]$- discount factor.

The transition between states is modeled by probabilities $P(s_{t+1} \mid s_t, a_t)$ that may be known or estimated.

Formulating the problem as an integer programming problem

The model is applied in pseudo-static or quasi-dynamic conditions (for example, in the context of planning in small time windows):

$$\min \sum_{i=1}^{N} \sum_{j=1}^{M} c_{ij} x_{ij}, \tag{2}$$

at restrictions:

- $\sum_j x_{ij} \leq 1, \forall i$- each task is assigned to no more than one performer.
- $\sum_i x_{ij} \leq C_j, \forall j$- do not exceed the performer's limit j;
- $x_{ij} \in 0,1$

where:

- c_{ij}- the cost of assigning a task i to an executor j;
- C_j- resource limitation.

Multi-agent production

Each executor is an agent that decides whether to accept or reject a task based on its local state stj. The system strives for equilibrium or global optimization through coordination (e.g., using auctions, RL, or contract algorithms):

Each worker is an agent that decides whether to accept or reject a task based on the local state s_t^j. The system strives for equilibrium or global optimization through coordination (e.g., auctions, RL, or contract algorithms):

$$\forall j \in \mathcal{A}: a_t^j = \arg\max_{a \in A^j} Q^j(s_t^j, a), \tag{3}$$

where:

Q^j- assessment of the benefit of the agent's action j;

$\mathcal{A}$- set agents (performers).

The following models are widely used to describe systems with dynamic task arrival and limited resources. In the context of IT services, tasks arrive at a certain rate (e.g. Poisson), and it is necessary to distribute them among several performers to minimize waiting time or SLA violations.

Mathematical model of task schedule

Let's consider a set of problems:

$$T = \{t_1, t_2, \dots, t_n\},$$

and many performers:

$$W = \{w_1, w_2, \dots, w_m\}.$$

Let c_{ij} be the costs (time, labor or cost) associated with the execution of the task t_i by the performer w_j, and $x_{ij} \in 0,1$ let be a binary variable that takes the value 1 if the task t_i is assigned to the performer w_j.

Target function:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} \cdot x_{ij}. \tag{4}$$

Restrictions:

Each task must be assigned to one performer:

$$\sum_{j=1}^{m} x_{ij} = 1, \forall i \in 1, \dots, n. \quad (5)$$

The total load of the performer must not exceed the permissible value L_j:

$$\sum_{i=1}^{n} x_{ij} \cdot d_i \leq L_j, \forall j \in 1, \dots, m, \quad (6)$$

where d_i is the duration of the task t_i.

Assignment model

This is a special case when the number of tasks and performers coincides, and each task must be assigned to one performer. Model is described next way:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} \cdot x_{ij}, \quad (7)$$

at restrictions:

$$\sum_{j=1}^{n} x_{ij} = 1, \sum_{i=1}^{n} x_{ij} = 1, x_{ij} \in 0,1. \quad (8)$$

RESULTS AND THEIR DISCUSSION

This paper addresses the problem of dynamic task allocation in an IT system, where multiple performers with limited resources require efficient optimization methods to enhance performance and minimize costs. The task is to optimally distribute tasks between performers under conditions of changing priorities, time and resource constraints.

The system must consider the following aspects:

- Dynamism: Tasks arrive randomly with different priorities, execution times, and resource requirements. These characteristics may change over time, requiring an adaptive approach to their distribution.
- Limited resources: Each executor has a limitation on time or computing resources (e.g. network bandwidth, server power).
- Task priorities: Tasks have different priorities depending on their importance, deadlines, and other factors, requiring flexibility in deciding which tasks to process first.
- Uncertainty: Incoming task statistics may change, and the exact information about tasks may not be complete at the time they are received.

Thus, the real-time task allocation problem can be formulated as: Minimize the total waiting time, the level of SLA violation (if any), and optimize resource utilization given the constraints.

Restrictions:

1. Each task must be assigned to one performer.
2. The total load of the performer must not exceed its resource limitations.
3. Priority tasks should be processed first.
4. The task execution time cannot exceed its time limit.

The dynamic nature of the problem suggests the use of adaptive and learning models that can change their decisions depending on changes in incoming data and the current state of the system.

Algorithm for solving the problem

The following diagram (Fig. 1) illustrates the algorithm for dynamic task distribution and strategy evaluation. It outlines the steps involved in selecting the appropriate distribution strategy, evaluating its effectiveness, and adapting to changing conditions using techniques such as MDP, multi-agent systems, and reinforcement learning.

The following table provides a quick overview of the various criteria that can be used to evaluate mathematical models in the task allocation problem in a dynamic IT environment. Each criterion plays a key role in deciding on the choice and application of a model in real-world conditions, where not only theoretical efficiency but also practical suitability is important.

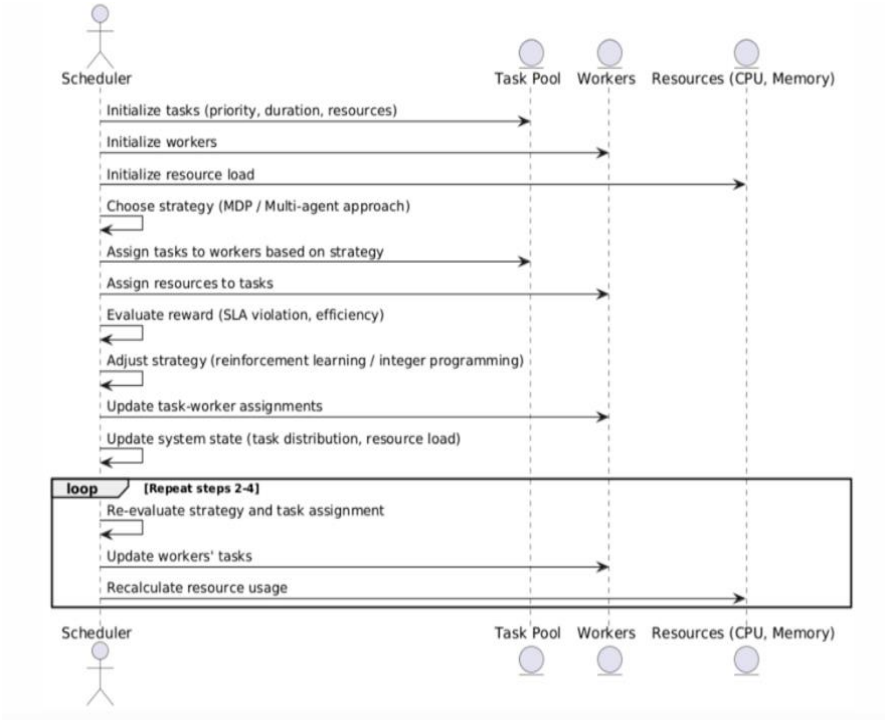

Figure 1. Task Distribution and Strategy Evaluation Algorithm

Note – compiled by the authors based on the main task-related objects.

Table 1. Criteria for evaluating task distribution models in a dynamic environment

Criterion	Description	Significance
Accuracy optimization	Measures how close the found solution is to the theoretically optimal one.	Important for assessing the effectiveness of models in solving optimization problems with constraints.
Time execution	Estimation of the duration of the problem-solving process for different sizes of input data.	Key criterion for tasks with real-time requirements.
Resilience to change in a dynamic environment	The ability of the model to adapt to changes in task parameters and changes in operating conditions.	Important for systems with frequent changes in conditions.
Flexibility and scalability	The ability of the model to work with different problem scales and adapt to different conditions.	Essential for working in environments with changing requirements or large volumes of data.
Solution quality when input data changes	Assessing the stability of a model's solution when input data such as cost or load change.	Important for ensuring stability and reliability of decisions in conditions of uncertainty.

End of table 1

Criterion	Description	Significance
Used resources	Consumption of computing resources (memory, CPU time, etc.) by the model.	Assessing the efficiency of resource use, especially in conditions of limited capacity.
Sensitivity to initial conditions	The influence of changes in initial conditions on the final solution.	Helps to assess the stability of the model and its ability to cope with data variations.
Explainability and transparency solutions	The ability to explain how the model came to its decision and understand its logic.	Key to trust in the system and to correct possible errors.
Security qualities service level agreement (SLA)	The ability of the model to meet service level agreements (e.g. response time).	Important to meet service standards and ensure required quality.
Simplicity implementations	Ease of integration of the model into an existing IT system and its adaptation to real-world conditions.	Assessing the complexity of implementing the model in an industrial environment.
Note – compiled by the author based on personal analysis and interpretation.		

If we apply these parameters in the context of your current project (for example, the task of assigning tasks between performers), we can use the table to compare different algorithms in more detail:

The input data for solving the task of optimal assignment of tasks to workers includes the following parameters:

1. Number of tasks (n_tasks) — the total number of tasks that need to be assigned to available workers.

2. Number of workers (n_workers) — the total number of workers available to perform the tasks.

3. Cost matrix (cost matrix) — a matrix representing the cost of assigning each task to a specific worker. The matrix has dimensions (n_tasks × n_workers), where each element represents the cost of assigning a particular task to a specific worker.

For example, if the number of tasks is 10 and the number of workers is 5, the following cost matrix might be generated:

$$\text{cost} = \begin{bmatrix} 47 & 13 & 97 & 79 & 83 \\ 34 & 89 & 26 & 74 & 31 \\ 56 & 43 & 29 & 8 & 69 \\ 34 & 13 & 60 & 98 & 44 \\ 48 & 65 & 81 & 17 & 74 \\ 81 & 54 & 74 & 90 & 37 \\ 98 & 66 & 40 & 95 & 76 \\ 35 & 58 & 94 & 91 & 73 \\ 85 & 75 & 42 & 14 & 93 \\ 89 & 6 & 8 & 59 & 28 \end{bmatrix}$$

Figure 2. Cost matrix

Note – compiled by the author

The values in this matrix define the cost of assigning each task to a specific worker. For instance, the first worker would complete the first task at a cost of 47 units, the second worker at

a cost of 13 units, and so on. This matrix serves as the basis for solving the optimal assignment task using various algorithms.

For each optimization algorithm (including linear programming, greedy algorithms, and random methods), this data is used to compute performance metrics such as:

Execution time — the time taken to solve the task.

Cost — the total cost of the solution compared to other methods.

Iterations (for iterative algorithms) — the number of steps required to reach a solution.

Memory — the amount of memory used to solve the task.

Standard deviation of task distribution — a measure of the spread of task assignments among workers, indicating the balance of the assignment.

Deviation from the optimal solution — the difference between the found solution and the theoretically optimal solution to the task.

Using these data points in combination with different algorithms allows for a comparative analysis of the solution methods based on the specified criteria.

```
MDP:
  Execution Time: 0.0003 seconds
  Total Cost: 503.6
  Memory Usage: 108.77 MB
--------------------------------------------------
Integer Programming:
  Execution Time: 0.0925 seconds
  Total Cost: 305.0
  Memory Usage: 110.30 MB
--------------------------------------------------
Multiagent:
  Execution Time: 0.0000 seconds
  Total Cost: 443
  Memory Usage: 110.30 MB
--------------------------------------------------
Scheduling:
  Execution Time: 0.0003 seconds
  Total Cost: 14.0
  Memory Usage: 110.34 MB
--------------------------------------------------
Assignment:
  Execution Time: 0.0003 seconds
  Total Cost: 113.0
  Memory Usage: 110.38 MB
--------------------------------------------------
```

Figure 3. Comparative Analysis of Solution Methods

Note – based on experimental results obtained by the author

The results (Fig.1) of the comparison between five optimization models — MDP (Markov Decision Process), Integer Programming, Multiagent, Scheduling, and Assignment — based on execution time, total cost, and memory usage are as follows:

1. Execution Time: The Multiagent and Scheduling models demonstrate the shortest execution time (0.0000 seconds), indicating high computational efficiency. The MDP and Assignment models also show fast execution times (0.0003 seconds), while Integer Programming exhibits the longest execution time (0.0925 seconds), reflecting its higher computational complexity.

2. Total Cost: The Scheduling model achieves the lowest total cost (14.0), making it the most cost-effective solution. The Integer Programming model provides a competitive cost (305.0), presenting a viable alternative for optimization. The MDP model incurs a higher cost (503.6), while the Assignment model results in a total cost of 113.0. The Multiagent model, although efficient in execution time, results in a higher total cost (443).

3. Memory Usage: Memory usage across all models is relatively consistent. The MDP model uses 108.77 MB, while the other models (Integer Programming, Multiagent, Scheduling, and Assignment) use between 110.30 MB and 110.38 MB.

CONCLUSION

The models analyzed in this study are especially relevant for solving dynamic and evolving optimization problems, which are commonly encountered in real-world applications such as resource allocation, scheduling, and decision-making in uncertain environments. Tasks with dynamic characteristics, such as those found in multi-agent systems, require continuous adaptation to changing conditions, making the choice of optimization model critical.

In dynamic environments, where parameters and conditions fluctuate over time, the Markov Decision Process (MDP) model proves particularly useful due to its ability to model sequential decision-making and account for future uncertainties. Similarly, multi-agent systems are well-suited for dynamic scenarios, as they allow for the decentralized coordination of agents that can independently respond to changes in the environment.

Integer Programming and Scheduling models, while more static in nature, can still be applied effectively when the dynamics of the task are well-defined or can be discretized, though they may require more computation to account for the real-time changes.

From the obtained results, we observe the following: the MDP model demonstrated the fastest execution time of 0.0003 seconds with a total cost of 503.6, but its memory usage was relatively high at 108.77 MB. The Integer Programming model, while taking slightly longer with an execution time of 0.0925 seconds, provided the lowest total cost of 305.0 and used 110.30 MB of memory. The Multiagent model exhibited a very low execution time of 0.0000 seconds, with a total cost of 443 and a memory usage of 110.30 MB, making it highly efficient in terms of time but less cost-effective. The Scheduling model had an execution time of 0.0003 seconds and a very low cost of 14.0, with minimal memory usage of 110.34 MB. The Assignment model had an execution time of 0.0003 seconds, with a total cost of 113.0 and a memory usage of 110.28 MB, demonstrating fast processing with moderate cost.

These findings underscore the importance of choosing an appropriate optimization model based on the dynamic nature of the task, where computational efficiency, adaptability to changes, and accuracy in solution quality are key factors to consider. The comparative analysis provides a valuable framework for selecting the best-suited model in the context of dynamic optimization problems, ensuring more efficient decision-making in complex, changing environments.

The results of this study demonstrate the effectiveness of various optimization models – MDP, Integer Programming, Multi-agent Systems, Scheduling, and Assignment — in solving dynamic, task-assignment problems. Despite their varying execution times and cost efficiencies, all models successfully addressed the problem within acceptable computational limits. Future research should focus on enhancing the scalability and adaptability of these models in real-time dynamic environments, incorporating machine learning techniques for better decision-making under uncertainty, and evaluating the robustness of these models when applied to larger-scale, complex scenarios. These advancements could significantly improve the practical applicability of these models in fields such as autonomous systems, supply chain management, and robotics.

CONFLICT OF INTEREST: The authors declare no conflict of interest.

FUNDING: no.

STATEMENT ON THE USE OF ARTIFICIAL INTELLIGENCE TECHNOLOGIES: The authors confirm that no generative artificial intelligence (AI) tools were used in the writing, editing, data analysis, or any other part of the preparation of this manuscript.

REFERENCES

Costa, A. R. da, París, D. M., Lujak, M., Rossetti, R. J. F., & Kokkinogenis, Z. (2024). Dynamic and cooperative multi-agent task allocation: Enhancing Nash equilibrium through learning.

In Proceedings of the 2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 1860–1866). IEEE. https://doi.org/10.1109/smc54092.2024.10831689

Chen, Y., Sun, Y., Wang, C., & Taleb, T. (2022). Dynamic task allocation and service migration in edge-cloud IoT system based on deep reinforcement learning. IEEE Internet of Things Journal, 9(18), 16742–16757. https://doi.org/10.1109/JIOT.2022.3164441

Gu, J., Chowdhury, M., Shin, K. G., Zhu, Y., Jeon, M., Qian, J., ... & Ghaffari, M. (2019). Tiresias: A GPU cluster manager for distributed deep learning. In Proceedings of the 16th USENIX Symposium on Networked Systems Design and Implementation (NSDI) (pp. 485–500). USENIX Association.

Han, Y., Li, X., & Zhou, Z. (2023). Dynamic task offloading and service migration optimization in edge networks. International Journal of Crowd Science, 7(1), 16–23. https://doi.org/10.26599/IJCS.2022.9100031

Joe-Wong, C., Sen, S., Lan, T., & Chiang, M. (2012). Multi-resource allocation: Fairness-efficiency tradeoffs in a unifying framework. In Proceedings of IEEE INFOCOM (pp. 1206–1214). IEEE.

Kunkler, M., & Rinderle-Ma, S. (2024). Online resource allocation to process tasks under uncertain resource availabilities. In Proceedings of the IEEE International Conference on Process Mining (ICPM) (pp. 137–144). IEEE. https://doi.org/10.1109/ICPM63005.2024.10723280

Lee, S.-J., Lee, H.-Y., & Ko, S. K. (2024). Stochastic programming method for volatile resource distribution problem. In Proceedings of the 15th International Conference on ICT Convergence (ICTC) (pp. 936–939). IEEE. https://doi.org/10.1109/ICTC62082.2024.10827280

Li, Y., Zhang, X., Zeng, T., Duan, J., Wu, C., Wu, D., & Chen, X. (2023). Task placement and resource allocation for edge machine learning: A GNN-based multi-agent reinforcement learning paradigm. IEEE Transactions on Parallel and Distributed Systems, 34(12), 3073–3089. https://doi.org/10.1109/TPDS.2023.3313779

Mao, H., Alizadeh, M., Menache, I., & Kandula, S. (2016). Resource management with deep reinforcement learning. In Proceedings of the 15th ACM Workshop on Hot Topics in Networks (HotNets '16) (pp. 50–56). ACM. https://doi.org/10.1145/3005745.3005750

Mohammad Asique E Rasool, Anoop Kumar and Asharul Islam, “Dynamic Task Offloading Optimization in Mobile Edge Computing Systems with Time-Varying Workloads Using Improved Particle Swarm Optimization” International Journal of Advanced Computer Science and Applications(IJACSA), 15(4), 2024.
http://dx.doi.org/10.14569/IJACSA.2024.01504122

Noghabi, S. A., Pan, X., Chien, S., Godfrey, P. B., & Schlesinger, C. (2020). Introducing Pause: A hybrid cluster scheduler for preemptible and non-preemptible workloads. In Proceedings of the 14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20) (pp. 973–990). USENIX Association.

Peng, Y., Bao, Y., Chen, Y., Wu, C., & Guo, C. (2018). Optimus: An efficient dynamic resource scheduler for deep learning clusters. In Proceedings of the Thirteenth EuroSys Conference (Article No. 1, pp. 1–14). ACM. https://doi.org/10.1145/3190508.3190517

Pradhan, S., Tripathy, S., & Matam, R. (2024). Towards optimal edge resource utilization: Predictive analytics and reinforcement learning for task offloading. Internet of Things, 26, 101147. https://doi.org/10.1016/j.iot.2024.101147

Yu, J., Lin, Z., Li, B., & Li, B. (2021). RL-Graph: A reinforcement learning approach to efficient multi-resource job scheduling. In Proceedings of IEEE INFOCOM 2021 (pp. 1–10). IEEE.

Zhao, L., Zhou, Y., & Huang, M. (2022). A survey of deep reinforcement learning for multi-agent systems. arXiv preprint arXiv:2201.11410. https://arxiv.org/abs/2201.11410

Авторлар туралы мәліметтер
Информация об авторах
Information about authors

Есмұхамедов Нұрғали Сейтқалиұлы – PhD докторанты, 3-курс, Интеллектуалды жүйелер кафедрасы, Халықаралық ақпараттық технологиялар университеті, Қазақстан
Есмухамедов Нургали Сейткалиевич — докторант 3-го курса, кафедра интеллектуальных систем, Международный университет информационных технологий, Казахстан
Yesmukhamedov Nurgali Seitkaliuly – 3rd year PhD student, Deportment of Intelligent systems, International Information Technologies University, Kazakhstan,
e-mail: nurgali.yesmukhamedov@gmail.com,
ORCID: 0009-0003-8772-9733

Сапақова Сая Заманбекқызы – физика-математика ғылымдарының кандидаты, доцент, Халықаралық ақпараттық технологиялар университеті, Қазақстан
Сапакова Сая Заманбековна — кандидат физико-математических наук, доцент, Международный университет информационных технологий, Казахстан
Saya Sapakova – Cand. of ph. and math. sc., Associate Professor, International University of Information Technology, Kazakhstan,
e-mail: s.sapakova@iitu.edu.kz,
ORCID: 0000-0001-6541-6806

Синчев Бахтгерей Құспанұлы – физика-математика ғылымдарының кандидаты, доцент, Халықаралық ақпараттық технологиялар университеті, Қазақстан
Синчев Бахтгерей Куспанович — кандидат физико-математических наук, доцент, Международный университет информационных технологий, Казахстан
Sinchev Bakhtgerey Kuspanovich – Cand. of ph. and math. sc., Associate Professor, International Information Technologies University, Kazakhstan,
e-mail: sinchev@mail.ru,
ORCID: 0000-0001-8557-8458

Есмухамедов Нұрмағанбет Сейтқалиұлы – PhD докторанты, 3-курс, Компьтерлік инженерия кафедрасы, Халықаралық ақпараттық технологиялар университеті, Қазақстан
Есмухамедов Нурмаганбет Сейткалиевич — докторант 3-го курса, кафедры Компьютерной инженерии, Международный университет информационных технологий, Казахстан
Yesmukhamedov Nurmaganbet Seitkaliuly – 3rd year PhD student, The Department of Computer Engineering, International Information Technologies University, Kazakhstan,
e-mail: yesmukhamedov.yeskendyr@gmail.com, ORCID: 0009-0006-0652-3082