Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 221 (2023) 41–48

Procedia Computer Science

www.elsevier.com/locate/procedia

10th International Conference on Information Technology and Quantitative Management

Multirobot Allocation In A Flexible Manufacturing System, Using Reinforcement Learning For Decision-Making, Case of Study

Gastón Lefranc*

*Pontificia Universidad Católica de Valparaíso, Zip Code 2430000, Chile,

Abstract

This paper presents a reinforcement learning approach for decision-making in assigning multiple robots in a flexible manufacturing system. The algorithm allows for learning the optimal action policy for assigning tasks to multiple robots, outperforming other methods. The case study involved three robots and four tasks, some requiring cooperation. The study demonstrates the effectiveness of the approach in comparison to other methods and the results indicate that the reinforcement learning approach increases production efficiency and reduces task completion time compared to traditional assignment methods.

Keywords: Multirobots; Allocation; FMS; Reinforcement Learning

1. Introduction

Flexible manufacturing systems (FMS) are used to have a more efficient and flexible production of products. In those systems, multiple robots are used to perform various tasks, such as material handling, assembly, and quality inspection. Some of these tasks are performed with two or more robots in cooperation. The allocation of various robots is a crucial issue in such systems, as it affects the overall production efficiency. Efficient task assignment to multiple robots can reduce the total time to complete tasks and consequently increase system productivity. Traditional methods for assigning multiple robots include centralized and decentralized approaches [1, 2].

The methods used for decision making are:

1.1 Centralized Assignment Method: A central controller assigns tasks to robots based on their capabilities, task requirements, and other factors. This method is useful for small systems and simple tasks [3].

1.2 Decentralized Assignment Method: Each robot has its own set of rules and decision-making algorithms to determine its task. This method is useful for complex tasks and large systems. Method involves distributed decision-making systems, where each robot makes decisions based on local information. [4].

* Corresponding author. Tel.: +56-990-794-152; fax: +56-990-494-152
E-mail address: gaston.lefranc@pucv.cl

1877-0509
10.1016/j.procs.2023.07.006

Nomenclature

FMS	Flexible Manufacturing Systems
RL	Reinforcement Learning
DM	Decision-Making x

1.3 Hybrid Assignment Method: The high-level tasks are assigned centrally. Robots assign lower-level tasks. This method combines the advantages of centralized and decentralized allocation methods [5].

1,4 Reinforcement learning (RL) method: It is based on experiences with robots and trial and error to determine the optimal way to assign tasks. It is useful for dynamic systems with frequently changing of tasks [6]. An agent interacts with an environment to achieve a certain goal. RL algorithms work by learning a policy that assigns states to actions, based on feedback from the environment in the form of rewards or penalties. RL has been successfully applied to various decision-making problems, including robot control, gaming, and resource allocation.

Assigning tasks to robots can be a complex decision-making problem due to task priorities, robot capabilities, and resource availability. The RL is a type of automatic learning, to make decisions in uncertain and complex environments.

When FMS is modeled as a Markov Decision Process (MDP), the Q-learning algorithm, RL algorithm, can be used to have an optimal task assignment policy. The MDP state is defined by the current task assignment and available resources, such as robot availability and task completion times. The robots could accept or reject a task, and the reward function is designed to encourage efficient and timely completion of the task and avoid resource conflicts [7].

To demonstrate the effectiveness of the approach, a scenario with three robots and four tasks is simulated. Each task has a priority level, and the robots have different capabilities such as speed and payload capacity. The simulation runs over multiple episodes, and the RL algorithm updates the Q values based on the rewards earned in each episode. The performance of the RL-based approach is compared to a reference approach that assigns tasks to robots based on their capabilities and priorities.

The best decision-making method depends on the specific system requirements. It is important to select a method that maximizes system efficiency and performance and minimizes implementation time and cost. In this article, a reinforcement learning approach is used for multi-robot allocation decision making in a flexible manufacturing system. The approach uses a deep reinforcement learning algorithm to learn the best action policy for assigning tasks to various robots. The results are compared to traditional assignment methods, showing that it outperforms them in terms of production efficiency and task completion time. The method uses a Q-learning technique to learn the best Q-valued function for each state-action pair.

|In this paper, it is proposed an RL-based approach to assign tasks to multiple robots in an FMS, using three robots and four tasks. The algorithm allows for learning the optimal action policy for assigning tasks to multiple robots, outperforming other methods. The case study involved three robots and four tasks, some requiring cooperation.

2. The reinforcement learning RL.

Reinforcement learning is a machine learning method that rewards desired behaviors and punishes unwanted behaviors. An agent learns through trial and error. The algorithm has three components: the agent, the environment, and the reward signal. Reinforcement learning can improve the efficiency and performance of an FMS, because it allows the controls of the robots to adapt to the environment and better decisions are made in the assignment of tasks.

There are disadvantages, such as computational complexity, due to the large amount of training data, difficulty in balancing exploration and exploitation, difficulty in interpreting the models and designing appropriate reward functions.

The reinforcement learning algorithm works in the following steps: The agent takes the state of the environment, including the available tasks, the capabilities of the robots, and the state of the system. The agent can assign a task to one of the robots. The environment responds to this action, the system updates its state and sends a reward signal to the agent, depending on the success or failure of its action. The agent updates its decision-making algorithm based on the reward signal and the new state of the environment. The agent repeats this process, continually learning and optimizing its decision-making algorithm.

Reinforcement learning can improve the efficiency and performance of an FMS by allowing robots to make better decisions about tasking multiple robots. The RL algorithm learns the best task assignment strategy considering the current state of the system and the reward function. The decision-making process is carried out by the RL algorithm itself, learning to optimize task allocation among multiple robots. It can adapt and adjust to changes in the environment, as well as changes in task requirements, robot capabilities, and system status. Reinforcement learning can reduce the need for manual intervention and increase automation in the system. [8, 9].

At each time step, the algorithm evaluates the system state, including task statuses and robot locations and availability, and selects an action based on its current policy function. The policy function maps states to actions and is learned over time through trial and error. After selecting an action, the RL algorithm observes the resulting system state and the reward based on task completion time. It updates its policy function using Q-learning or policy gradients. This process repeats over multiple time steps until all tasks are completed.

In summary, the decision-making process in this example is carried out by the RL algorithm, which learns the best task allocation strategy based on the current state of the system and the reward function through trial and error [10].

3. Case of Study.

3.1 Petri Net

Suppose FMS produces a product that welds two parts and then assembles it with a third, using different robots. Petri Nets are shown in fig. 1. In the Petri Net, the Places represent a queue of tasks, and the Transitions represent the assignment of the task to one of the available robots (Table 1 and 2).

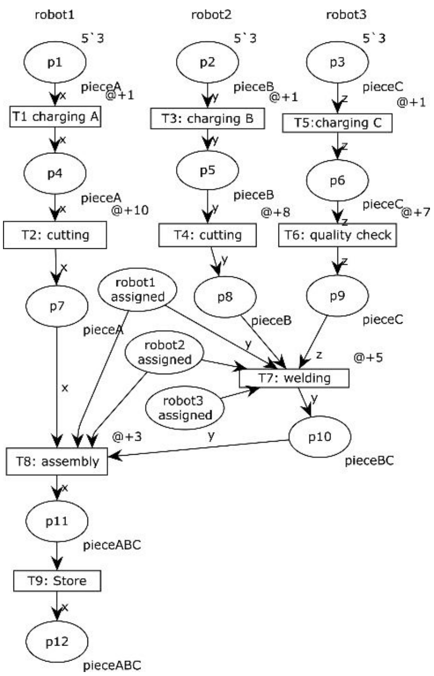

Figure 1 Petri Net of FMS proposed.

1. Load part A using robot 1.
2. Load part B using robot 2.
3. Load part C using robot 3.
4. Cut part A using robot 2.
5. Cut part B using robot 3.
6. Quality check part C
7. Weld part B and part C, with robot 2 and robot 3.
8. Assemble parts A, B+C, robot1 and robot 2.
9. Store A+B+C

Table 1 Places

P1	robot1 waiting
P2	robot2 waiting
P3	robot3 waiting
P4	ready to cut A robot1
P5	ready to cut B robot2
P6	ready to quality control
P7	piece A ready robot1
P8	piece B ready robot2
P9	piece C ready robot3
P10	B+C welded robot2 y 3
P11	A+(B+C) Asembled r1,2
P12	(A+B+C) Store robot1

Table 2 Transition

T1	Load piece A
T2	Cutting piece, A
T3	Load piece B
T4	Cutting piece B
T5	Load piece C
T6	quality check piece C
T7	Welding B+C
T8	Assembly A+(B+C)
T9	Store A+(B+C)

Arcs connect places and transitions indicating the flow of tasks and resources through the system. Task assignment to robots can be modeled by using additional places to represent the state of each robot and transitions to represent decision making and task performance by each robot [11, 12].

Petri nets are used to simulate the proposed methods, since it allows modeling dynamic systems with concurrent processes. The mapping of robots, machines, and other elements of the Flexible Manufacturing System (FMS) is done with these networks. Tasking robots is treated separately from other resources because robots have unique capabilities.

and characteristics, such as the ability to move, technical skills, and the payload they can support. This assignment requires precise planning and coordination for the efficient and synchronized execution of the robot's tasks.

The Petri net model may not fully address the structural conflicts described in steps 1 through 9 and the proper release of the robots after completing the actions. These omissions are the result of simplifications made in this work and should be considered as limitations for future research.

A task of loading, cutting and quality control, respectively, is assigned to the robots R1, R2, R3, in parallel. To carry out welding, they can be assigned in pairs of robots (Robot 1 and Robot 2, Robot 2 and Robot 3, or Robot 1 and Robot 3), according to their availability. Pairs of robots are also assigned for assembly.

The task assignment decision making depends on the method used, the complexity of the tasks, the capabilities of the robots, the size of the system and the time constraints. Petri Nets allow modeling the system and the possible assignment sequences, optimizing the assignment of tasks to the robots, improving the efficiency and performance of the flexible manufacturing system [1, 13].

3.1. Allocation using a centralized allocation method.

The centralized method assigns tasks to available robots from a central administration. The administrator considers the availability of the robot, the skill required for the task, and the location of the robot, to make the decision to assign a given task and in what order. This method can help ensure proper resource utilization and work efficiency. At the same time, it is less flexible than other allocation methods and requires constant monitoring and maintenance by the administrator. This method is effective for simple systems. In more complex systems it is complicated because the algorithms imply high computational costs [1].

The central controller assigns a certain task to each robot and coordinates the execution of those tasks according to a certain plan (Figure 2). The single controller would manage all the actions and decide which task to perform at which time. The efficiency of the controller would determine the time of each task. Table 3 shows the time of the robots involved. In a centralized robot and task assignment system, a central administrator oversees all available robots and tasks in the system.

The single controller would manage all the actions and decide which task to perform at which time. The efficiency of the controller would determine the timing of each task. Tables 3 displays the time for robots involved.

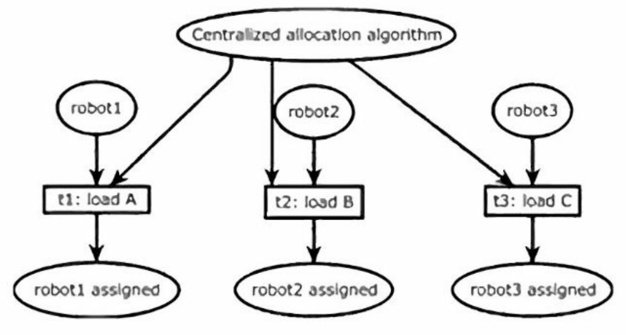

Figure 2 Centralized allocation method

Table 3: Time matrix for robots

Robots	Load A	Load B	Load C	Cut A	Cut B	Q control C
Robot 1	20			40		
Robot 2		15			30	
Robot 3			15			30

Robots	Welding B and C	Assembly A to B+C	Store A+B+C
Robot 1		50	35
Robot 2	30	50	
Robot 3	30		

Average time per task 39,4 minutes and Total production time 265 min

In a centralized robot and task assignment system, a central administrator oversees all available robots and tasks in the system.

Decision making in a centralized robot and task assignment set depends on the algorithm or rule used to assign robots to tasks. For example, if the algorithm prioritizes work efficiency, robots can be assigned to tasks that are completed quickly, instead of considering other variables such as the importance of the task or the experience of the robot in the task at hand.

The centralized robot and task assignment method has the advantage of providing centralized control over the assignment process, which can help ensure optimal resource utilization and work efficiency. However, it can also be less flexible than other allocation methods and may require constant monitoring and maintenance by the administrator.

3.2. Allocation using a hybrid allocation method.

In a hybrid assignment method, decision making is between the central controller and the controllers of each robot. The central controller sends the task sequence, where the robot controllers make the decision, in real time, to perform the task or not. This approach is more flexible [1].

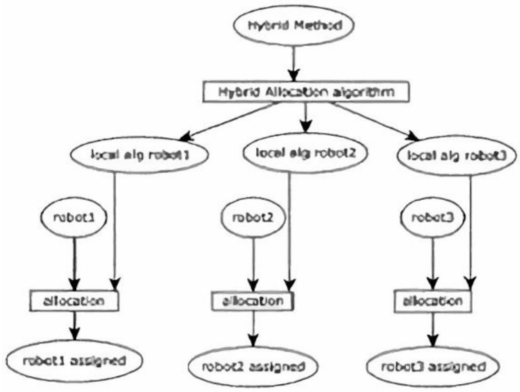

Fig. 3 Hybrid allocation method

Table 4: Time matrix for robots

Robots	Load A	Load B	Load C	Cut A	Cut B	Q control C
Robot 1	18			35		
Robot 2		12			28	
Robot 3			12			30

Robots	Welding B and C	Assembly A to B+C	Store A+B+C
Robot 1		45	25
Robot 2	45	45	
Robot 3	45		

Average time per task 27,8 min and Total production time 250 min.

One implementation of the hybrid method is to have the central controller assign centralized high-level tasks to each robot, while the robots themselves decide how to assign lower-level tasks. Both centralized and decentralized methods are used to assign tasks to robots in this approach.

For example, the centralized algorithm could assign tasks to robots according to their ability and location, while the robot controller can decide the best way to perform the task based on knowledge of the environment.

Decision making in a hybrid robot and task assignment system can be more complex than in purely centralized or decentralized systems, as multiple factors must be considered and centralized and decentralized decisions must be balanced. The centralized manager must set the rules and boundaries for decentralized decision making and ensure consistent and effective decisions.

With the hybrid assignment method, it has been possible to optimize the total production time by assigning tasks to the most efficient robots in terms of time. Figure 3 shows the Petri Net, Table 4 presents the time of the robots.

With this hybrid method, each robot can make decisions regarding lower-level tasks, such as selecting specific raw materials to handle or which parts to inspect; and combines centralized control and decentralized decision-making.

3.3. Allocation using a Reinforcement learning (RL) allocation method.

The RL method is a machine learning approach that can be used for robot and task assignment. This approach involves training an RL agent to make optimal decisions in real-time about how to complete each task by receiving feedback from the environment. The RL agent uses Q-learning, to learn which tasks to assign to which robots. Each task has a reward function that reflects the robot's performance on the task. The RL algorithm learns to select the optimal tasks based on the reward function.

In an RL-based robot and task assignment system, the RL agent uses a reinforcement learning algorithm to learn how to assign tasks to robots efficiently. The agent receives information about the environment, such as the location and status of robots and tasks and uses this information to make allocation decisions.

The method permits an efficient task assignment to robots, improving production time and resource utilization [14]. An RL algorithm, such as Q-learning, is performed so that the robot learns to select the optimal tasks based on the reward function. Fig. 4, shows Reinforcement learning allocation method. Table 5 shows the time required per robot in the Petri Net.

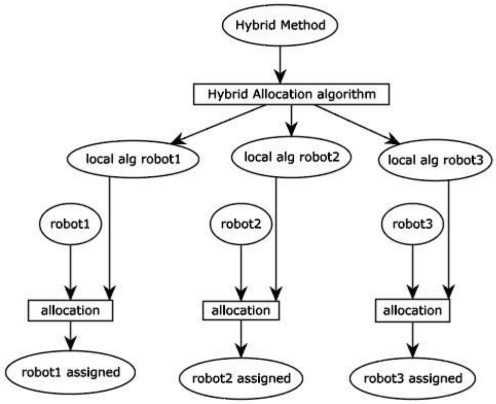

Fig. 4 Reinforcement learning allocation method

Table 5: Time matrix for robots:

Robots	Load A	Load B	Load C	Cut A	Cut B	Q control C
Robot 1	15			20		
Robot 2		10			25	
Robot 3			10			30

Robots	Welding B and C	Assembly A to B+C	Store A+B+C
Robot 1		40	25
Robot 2	40	40	
Robot 3	40		

Average time per task is 23.9 min and Total Production Time is 215 min.

The mathematical model of reinforcement learning that assigns tasks to an FMS with three robots and four tasks can be formulated using the Markov Decision Process (MDP) framework, which has a set of states, S; a set of actions, A; a transition function, P(s,a,s') and a reward function, R(s,a) [15, 16].

In order to have efficient cooperation between robots in a cooperative task, the RL method assigns the task to the most suitable and available pairs of robots, taking the different combinations of robots that can cooperate in the task. A reward function is established to evaluate the performance of the set of robots and the quality of their collaboration. The reward feature and actions may vary by app and need to be carefully designed. The RL agent continually evaluates the rewards and penalties received based on its decisions, in order to assign the correct robot to perform a task. The RL agent adjusts its behavior based on feedback from the environment, allowing it to be adaptive. Using this RL allocation method, it has been possible to optimize the total production time by assigning tasks to the most efficient robots in terms of time.

The RL-based robot and tasking approach can offer greater efficiency and adaptability, as the RL agent learns to assign tasks based on feedback received from the environment. To ensure optimal decision making, this method can be more complex to do and requires good training and continuous evaluation [17].

4. The results.

The methods are implemented with CPN 5 Tools software. CPN Tools allows for the graphical representation of the system components, their states, and their interactions within a Colored Petri Net (CPN) framework.

The CPN model to represent the decision-making entity of the centralized method, the central coordinator is defined by a place, with the rules and logic for task assignment using variables, guards, and conditions. The central coordinator interact with the robots are modeled by transitions and arcs that represent communication and task assignment.

In the CPN model of the hybrid method, a central coordination place and places are defined for each robot. Decision making has rules and a logic, specified for the central coordinator and for each robot. These decisions are made with interactions and coordination between the central coordinator and the robots. The decision making of the robots is autonomous.

The CPN model of the RL-based method incorporates agents and algorithms. The agents and the decision-making process are represented by places and transitions. Within agents, states, actions, rewards, and the learning mechanism are specified by CPN constructs. The interactions between RL agents, with the environment and with the tasks to be assigned, the interactions between the RL agents, the environment, and the tasks to be assigned. The CPN simulation allows to see the learning process and decision making of RL agents, and to evaluate the performance of the RL based task allocation.

The simulation results highlight the significant advantages the RL-based approach over the baseline approach the other methods, in terms of task completion time and resource utilization. The RL-based approach exhibits remarkable adaptability to changing task priorities and robot availability, enabling efficient and timely task completion.

The allocation sequences derived from the simulation demonstrate the effectiveness of simultaneous allocation of Robot 1, Robot 2, and Robot 3 for loading, cutting, and quality control tasks. For welding and assembly, the allocation can be optimized by assigning different robot combinations, such as Robot 1 and Robot 2, Robot 2 and Robot 3, or Robot 1 and Robot 3. Storage tasks can be efficiently performed by any available robot.

Table 6 provides a comprehensive comparison of execution times and average task completion times for the three methods, clearly demonstrating the superiority of the RL-based approach. It significantly outperforms centralized and hybrid methods, exhibiting the shortest execution time and average task completion time. In particular, the RL-based approach may require initial time to explore and learn from interactions with the environment.

In the simulation, the RL method achieves a total execution time of 215 minutes, with an average time per task of 23.9 minutes. In contrast, the baseline approach initially requires a total production time of 280 minutes, with an average time per task of 31.1 minutes. These results highlight the exceptional efficiency of the RL-based approach, significantly reducing both overall production time and average task completion time. RL algorithms require time to explore and learn from interaction with the environment before making optimal decisions.

Table 6: Time comparison of Allocation Methods

Comparison	Centralized Method	Hybrid Method	RL Met hod
Task	min	min	min
Task 1: Load part A by robot 1	20	18	15
Task 2: Load part B by robot 2	15	12	10
Task 3: Load part C by robot 3	15	12	10
Task 4: Cut part A	40	35	20
Task 5: Cut part B	30	28	25
Task 6: Quality control part C	30	30	30
Task 7: Weld part B to part C	30	45	40
Task 8: Assembling the welded parts with part A	50	45	40
Task 9: Storage	35	25	25
Average time per task	39,4	27,8	23,9
Total production time	265	250	215

Each allocation method has its own advantages and disadvantages. The centralized method performs well in simple systems but becomes increasingly slower as complexity grows. The decision making for the assignment of tasks, the hybrid method offers greater flexibility and potentially shorter execution times, albeit at the cost of potential coordination errors and increased complexity. The RL-based approach stands out as the most flexible and adaptable, as the RL agent learns to make optimal decisions in real-time based on the environment and feedback.

Comparing the quantitative results with others research is challenging. However, qualitatively, similar conclusions the fhave been reached that employ RL methods. For example, Bahrpeyma F, et al., present a review of multi-agent reinforcement learning applications in smart factories, focusing on scheduling and transportation tasks. Different characteristics of Multi agent RL are analyzed special the ability to handle uncertainty in decentralized systems and self-organization [18]. Jingyuan Lei, Jet al, presents an RL algorithm for making operational decisions based on real-time data, showing improved performance compared to the centralized methods in Industry 4.0 SF. The algorithm autonomously assigns production and logistics tasks in SF. This dynamic task assignment model used an objective function [19].

Overall, the simulation results emphasize the superiority of the RL-based approach in terms of efficiency, adaptability, and resource utilization compared to reference methods. The findings contribute significantly to the field by showing the RL-based approach as a highly effective solution for task assignment and coordination in multi-robot systems.

The time it takes to make decisions is important for evaluating system performance and optimizing task assignment. To maximize the efficiency of the system, one must know the durations of the tasks and their impact on planning and execution.

The Petri net in this work is a simplified representation of the system, adapted to the simulation to facilitate analysis and understanding. Certain simplifications focus on assigning and releasing robots upon task completion, which may not be explicitly reflected in the model. The release of the robots is essential for their reallocation and flexibility in the system.

If there is an initial marking, it implies a pre-assignment of robots to tasks and to manufacture a certain number of pieces. This pre assignment may be a constraint on this job. This paper does not address rescheduling, and it is estimated that it will be necessary to do so in the future to handle unconsidered system changes or disturbances.

Conclusions

In this paper, the assignment of robots and tasks has been presented, in a flexible manufacturing system, using the RL reinforcement learning approach for decision making for the assignment of multiple robots.

This problem can be critical as it affects production efficiency. A deep reinforcement learning algorithm is used to learn the optimal action policy for assigning tasks to multiple robots. A case study is presented to illustrate the effectiveness of the proposed approach and it is compared with centralized method and hybrid method for the assignment of robots and tasks. The case study considers a system with three robots and four tasks. The results show that RL approach outperforms traditional assignment methods in production efficiency and task time.

The results are obtained from the simulation with Petri nets, using CNP Tools. The table comparing execution times and average task completion times for the three methods, showing that the RL-based approach has the shortest execution time and average task completion time. However, the RL-based approach may initially take more time to explore and learn from interaction with the environment before making optimal decisions.

The RL method is the most flexible and adaptable of the three, as the RL agent can learn to make optimal decisions in real time based on the environment and feedback, which can result in shorter execution times than the other methods. The aim is to develop an effective information system that integrates computerized versions of the algorithms as a practical outcome of the research. The incremental approach [20, 21] to develop such a system is currently envisaged.

References

[1] Khamis, A., Hussein A., Elmogy A. M., (2015) Chapter: Multi-robot Task Allocation: A Review of the State-of-the-Art, In book: Cooperative Robots and Sensor Networks 2015: Springer International Publishing. DOI: 10.1007/978-3-319-18299-5_2. Editors: Anis Koubâa, J.Ramiro Martínez-de Dios.

[2] Fazal N, Khan MT, Anwar S, Iqbal J, Khan S. Task allocation in multi-robot system using resource sharing with dynamic threshold approach. PLoS One. 2022 May 4;17(5):e0267982. doi: 10.1371/journal.pone.0267982. PMID: 35507628; PMCID: PMC9067701.

[3] Zhao, D.; Yang, C.; Zhang, T.; Yang, J.; Hiroshi, Y. A Task Allocation Approach of Multi-Heterogeneous Robot System for Elderly Care. Machines 2022, 10, 622.https://doi.org/10.3390/machines10080622

[4] H. Hanna, "Decentralized approach for multi-robot task allocation problem with uncertain task execution," *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, AB, Canada, 2005, pp. 535-540, doi: 10.1109/IROS.2005.1545037.

[5] G. Li *et al*., "Hybrid dynamical moving task allocation methodology for distributed multi-robot coordination system," *2015 IEEE International Conference on Mechatronics and Automation (ICMA)*, Beijing, China, 2015, pp. 1412-1417, doi: 10.1109/ICMA.2015.7237692.

[6] Elfakharany A., Yusof R. and Ismail Z. (2019). “Towards Multi Robot Task Allocation and Navigation using Deep Reinforcement Learning,” Journal of Physics: Conference Series 1447 (2020) 012045, doi:10.1088/1742-6596/1447/1/012045.

[7] Kober J., Bagnell J.A., Peters J., (2013) “Reinforcement Learning in Robotics: A Survey”. The International Journal of Robotics Research 32(11):1238-1274, DOI: 10.1177/0278364913495721

[8] AlMahamid F., Grolinger K.(2021), Reinforcement Learning Algorithms: An Overview and Classification. 2021 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), 2021, pp. 1-7. DOI: https://doi.org/10.1109/CCECE53047.2021.9569056

[9] Zhenping Li, Wenyu Li (2015), “Mathematical Model and Algorithm for the Task Allocation Problem of Robots in the Smart Warehouse”. American Journal of Operations Research 05(06):493-502, DOI: 10.4236/ajor.2015.56038

[10] Park B, Kang C, Choi J. Cooperative Multi-Robot Task Allocation with Reinforcement Learning. *Applied Sciences*. 2022; 12(1):272. https://doi.org/10.3390/app12010272

[11] Jensen K., Kristensen L. M. Coloured Petri Nets Modelling and Validation of Concurrent Systems (2009). Springer-Verlag Berlin Heidelberg ISBN 978-3-642-00283-0, e-ISBN 978-3-642-00284-7. DOI 10.1007/b95112

[12] Xu, G.; Chen, Y., Petri-Net-Based Scheduling of Flexible Manufacturing Systems Using an Estimate Function.Symmetry 2022, 14, 1052.https://doi.org/10.3390/sym14051052

[13] Yahia Zare Mehrjerdi, A decision-making model for flexible manufacturing system (2009). Assembly Automation 29(1):32-40, DOI: 10.1108/01445150910929839

[14] Orr, J.; Dutta, A. MultiAgent Deep Reinforcement Learning for Multi-Robot Applications: A Survey. Sensors 2023, 23, 3625. https://doi.org/10.3390/s23073625

[15] Kober J., Bagnell J.A., Peters J., Reinforcement Learning in Robotics: A Survey(2013). The International Journal of Robotics Research 32(11):1238-1274. DOI: 10.1177/0278364913495721

[16] Maurya A. K., Jayswal S. C., Reinforcement Learning in Robotics: A Survey (2013) IJSRD - International Journal for Scientific Research &Development| Vol. 3, Issue 04, 2015 | ISSN (online): 2321-0613

[17] Huang B., Lu, X. S., Abusorrah A., Scheduling of Resource Allocation Systems with Timed Petri Nets: A Survey. ACM Computing Surveys Volume 55Issue 11Article No.: 230pp 1–27https://doi.org/10.1145/3570326

[18] Bahrpeyma F, Reichelt D. A review of the applications of multi-agent reinforcement learning in smart factories. Front Robot AI. (2022) Dec 1;9:1027340. doi: 10.3389/frobt.2022.1027340. PMID: 36530498; PMCID: PMC9751367.

[19] Jingyuan Lei, Jizhuang Hui, Fengtian Chang, Salim Dassari & Kai Ding (2022) Reinforcement learning-based dynamic production-logistics-integrated tasks allocation in smart factories, International Journal of Production Research, DOI: 10.1080/00207543.2022.2142314

[20] Alter, S. Defining information systems as work systems: implications for the IS field. *Eur J Inf Syst* 17, 448–469 (2008). https://doi.org/10.1057/ejis.2008.37

[21] Filip F.G. Decision-making perspective for designing and building information systems. INT.J COMPUT COMMUN-IJCCC, ISSN 1841-9836, Vol.7 (2012), No.2 (June), pp. 264-272.