Proceedings of the Canadian Society for Mechanical Engineering International Congress
32nd Annual Conference of the Computational Fluid Dynamics Society of Canada
Canadian Society of Rheology Symposium
CSME/CFDSC/CSR 2025
May 25–28, 2025, Montréal, Québec, Canada

A Fuzzy Inference-Based Architecture for Allocating Multi-Robot Tasks

Matthew Tidd[1*], Rickey Dubay[1], Hung Cao[2]

[1]Department of Mechanical Engineering, University of New Brunswick, Fredericton NB, Canada
[2]Faculty of Computer Science, University of New Brunswick, Fredericton NB, Canada
[*]e-mail: mtidd2@unb.ca

***Abstract*—Multi-robot systems (MRS) are becoming more popular due to advancements in artificial intelligence (AI), as well as their ability to resolve task complexity. One of the most challenging problems with the use of MRS is the allocation of tasks to robots. This paper proposes a methodology for structuring bid formulation within distributed, auction-based *Multi-Robot Task Allocation* (MRTA). A fuzzy inference system (FIS) is proposed as a means of evaluating the suitability of robots for a given task by considering objective factors such as their load history, distance to the task, and the total distance travelled. The suitability of a robot is combined with its capability and put forth as a bid in the auctioning process. An adaptive neuro-fuzzy inference system (ANFIS) was developed as a custom implementation within Keras to extract a higher dimensional relationship between the input variables and to leverage parallelized computation. The previously designed FIS was used as a data generator, and the ANFIS was trained to perform regression between the objective parameters of a robot and its suitability. The results from this were a 9.8x decrease in inference time, down from 0.315 seconds to 0.032 seconds. Future work involves exploring using an artificial neural network (ANN) to better approximate the FIS while maintaining the parallelized computation, as well as deploying and evaluating the methodology in simulation and on real robotic agents.

***Keywords*—Multi-Robot Systems; Task Allocation; Mobile Robotics; Fuzzy Logic; Machine Learning**

I. INTRODUCTION

Mobile robotics have become more popular in recent years due to several key technological advancements, namely the rampant development and implementation of artificial intelligence (AI). This, in turn, has resulted in their demand increasing dramatically [1]. With the technological capabilities of robots increasing, the complexity of their use cases also increases. To counter this rise in complexity, researchers have turned towards using multi-robot systems (MRS), creating one of the most extensive research domains in robotics [2]. An MRS consists of a group of robots that have been designed so that they may achieve or perform some collective behaviour or task. Deployment of a single robot to perform complicated tasks is often time-consuming and energy exhaustive, and many robots may be required to accomplish complex or dangerous tasks.

MRS have seen numerous applications within a variety of fields such as manufacturing, construction, mining, inspection, warehouse scheduling, exploration, and agriculture, among others [3], [4]. The benefits of MRS include resolving task complexity, decreased task completion time, and increased reliability and overall simplicity in design [5]. MRS can therefore be used to perform complex tasks faster than a single robot, and do so in a distributed manner. This inherently imbues them with an innate sense of flexibility and robustness.

One of the most challenging problems relating to the use of MRS is the optimal assignment of tasks to robots in a manner that optimizes system performance subject to some set of constraints. Within the literature, this has been described as the *Multi-Robot Task Allocation* problem (MRTA), a classic topic of research for several years within the robotics community. The MRTA problem seeks to address the difficulty associated with finding which task should be assigned to which robot within an MRS, such that the overall system goals may be achieved. This process is further abstracted when dealing with heterogeneous robots equipped with different capabilities, as well as when the tasks must be assigned in real-time as they appear within the environment [5], [6].

Coordinating teams of heterogeneous robots each equipped with different sensory and locomotive capabilities therefore proves to be quite a challenging endeavour, requiring the use of a robust yet flexible system to model, coordinate, and exchange robot skills and behaviours [7]. Existing methodologies can be broken down into two important main groups: auction- and optimization-based approaches [5]. While many of these strategies have been implemented within the literature, they both lack validation through real-time experimentation, as they are often implemented in simulation only. Furthermore, these strategies suffer from difficulties in formulating multi-objective functions to adequately capture the design requirements due to the complexity of the search space [2], [3], [5]. As such, within recent years, there has been interest in alternative methods for formulating the MRTA problem, such as the use of fuzzy logic as a method of task allocation, or the use of machine learning (ML) techniques such as artificial neural networks (ANNs) [6], [8]–[11].

Considering these points, this paper serves as a preliminary investigation into an alternative methodology for bid formulation within auction-based task allocation. This research specifically examines the usage of a fuzzy inference system (FIS) to infer a robot's suitability for a given task based on a set of objective parameters that are queried from the robot during task allocation. An adaptive neuro-fuzzy inference system (ANFIS) is examined as a means to augment this approach by improving allocation time through leveraging parallelized computation, allowing for faster inference times.

II. PROBLEM FORMULATION

This research has been performed under the assumption that robots can locate themselves within the environment by using localization techniques. It is assumed that a framework like the Nav2 stack for ROS2 is used to manage each robot's path planning and navigation. The focus of this work is therefore on the dynamic nature of task allocation, where tasks emerge over time after robots have already been deployed within an environment. No prior information is available regarding the requirements of the task, its location, or its time of appearance. All robots are informed of the location of a task within the environment once it appears.

A task can be defined by both its position within space and its requirements. For a task to be considered complete, a robot with a capability that matches the requirement of the task must arrive at the task site. It is assumed that tasks can have more than one requirement, which would therefore require more than one robot for completion. Conversely, a robot is defined by its position within space, its capability, and a set of objective parameters that contextualize it within the task-allocation framework. These objective parameters are the load history of the robot, the distance that the robot is from the current task, as well as the distance history of the robot, which represents the total distance travelled by the robot thus far.

Figure 1 illustrates a sample scenario of four robots and one task. For simplicity, it is assumed that this scenario is a simple industrial inspection task, but the proposed methodology is not limited to such. In this scenario, robots can have either one of two capabilities, imagery or measurement, meaning that some robots are responsible for taking images and some are responsible for collecting measurements. As such, tasks can either require an imagery robot, a measurement robot, or both. This requirement is not known to the robots beforehand. This problem can be considered as an instance of the ST-MR-IA sub problem [12], as each robot can complete at most one task at a time, tasks can require more than one robot for completion, and there is no prior knowledge about the scheduling of tasks.

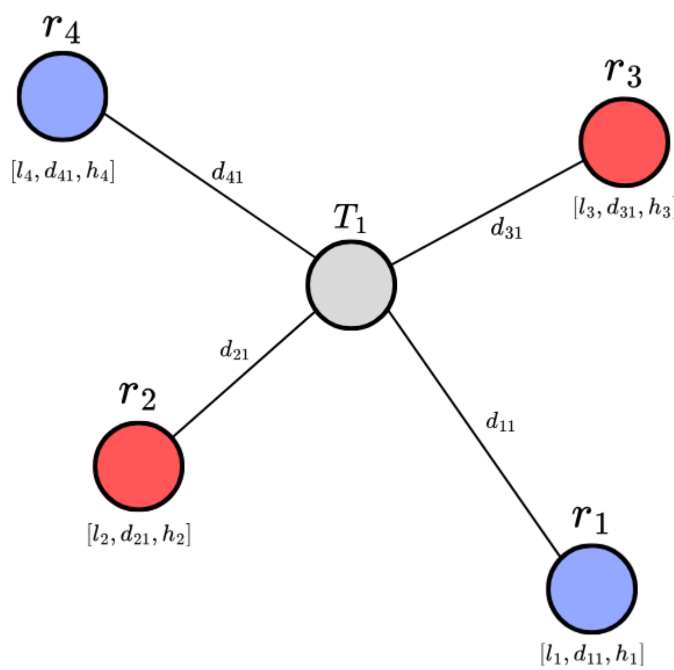

Figure. 1. Task allocation sample scenario with two distinct robot classes.

Let it be assumed that blue robots represent those with the imagery capability, and red robots represent those with the measurement capability. Beneath each robot is a vector that contains the objective parameters describing a given robot r_i. The value d_{ij} represents the distance that robot i must travel to get to task j, and the values of l_i and h_i represent the load history and the distance history of a given robot, respectively. The goal of this research is therefore to design a bid formulation strategy for distributed, auction-based task allocation that amalgamates the objective parameters of a given robot into a bid that is representative of their suitability for a given task. More specifically, this research seeks to develop a methodology that:

1) Correctly allocates robots to tasks, matching capabilities to requirements,
2) Formulates bids without the usage of an objective or cost function,
3) Distributes tasks evenly among agents within the MRS, and
4) Minimizes the sum of distance travelled across agents in the MRS.

III. PROPOSED METHODOLOGY

To encapsulate both a robot's suitability for a given task based on its objective parameters (load history, distance to the task, and distance travelled), as well as its capability for the task at hand (the tasks that it is able to complete), this research proposes a bid given by Equation 1.

$$Bid = \begin{bmatrix} Capability \\ Suitability \end{bmatrix} \quad (1)$$

A robot's *capability* is based on the robot's type, i.e. what it was designed to achieve, and is required to ensure a strict matching between task requirement and robot type. This ensures that for every task T, there is a robot with a capability that matches the task requirement. In terms of the example scenario presented within Figure 1, the available capabilities are *Imagery* or *Measurement*.

Similarly, the value of *suitability* is a scalar metric that defines how suitable a given robot is based on its objective parameters and is inferred through the use of a fuzzy inference system. The FIS is used to combine the objective parameters into a single scalar metric and therefore factors in the inter-connectivity between objective parameters and their overall contributions to how suitable a robot is for a given task. The following objective parameters were selected, and as these objective parameters increase in value, a robot's suitability therefore decreases:

a) Load history was selected to distribute the tasks evenly among robots.
b) Distance to the task was chosen to prioritize robots that are closer to the task, to help minimize the total distance travelled by members of the MRS.
c) Finally, a distance history was chosen to also help in distributing tasks.
   If two robots have similar load histories and distances to the task, the task should be assigned to the less travelled robot.

A. FIS Design

A conventional Type 1 Mamdani FIS was designed for the task allocation scenario. Fundamentally, a fuzzy inference system represents a static nonlinear mapping between its inputs and outputs, where crisp (non-fuzzy) inputs are mapped to crisp outputs using fuzzy logic as the mapping modality. The inputs are typically represented as *linguistic variables*, each subdivided into *linguistic values* based on their degree of membership, such that they can be used within rules for reasoning. Within the context of the task allocation scenario presented in Figure 1, the input linguistic variables to the FIS were selected to be the objective parameters defined previously: distance to the task, load history, and distance history, while the output represented the suitability of a robot.

There are five important components within a FIS:

1) The *rule-base*, where the rules used in the mapping of inputs to outputs are held,
2) The *data-base*, which contains the membership functions of the fuzzy sets that are used within the rules,
3) A *fuzzification interface*, where crisp inputs are converted to fuzzy sets to be used by the inference mechanism,
4) An *inference mechanism*, where expert-level decision making is emulated, and
5) A *defuzzification interface*, where the fuzzified output of the inference mechanism is turned back into a crisp output.

The fuzzy inference system is therefore governed by two important components: the **membership functions** as well as the **rule-base**.

1) Membership Function Design: Each of the input linguistic variables is associated with a set of membership functions, which define the degree to which an input belongs to different linguistic values. These membership functions are used within the fuzzification process, where crisp input values are mapped to fuzzy sets. The design of these functions therefore directly influences the reasoning and decision-making capabilities of the system. The input variables can be represented as the following:

a) *load history*, denoted by x_1,
b) *distance from the task*, denoted by x_2, and
c) *distance history*, denoted by x_3.

For each input variable, the linguistic values *low (L)*, *medium (M)*, and *high (H)* were selected. The output of the FIS is the linguistic variable *suitability*, denoted by y, and the linguistic values *very low (VL)* and *very high (VH)* were selected in addition to the previously mentioned three values for the output. The membership functions for each of these linguistic values, for both the input and output variables, were chosen to be triangular membership functions.

Triangular membership functions were employed due to their simplicity, computational efficiency, and effectiveness in representing gradual shifts in membership as the input values change. These membership functions are defined by the parameters a, b, and c, and have a form given by Equation 2.

$$\mu(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & c \leq x \end{cases} \tag{2}$$

This therefore meant that the parameters a, b, and c had to be chosen for each membership function corresponding to each linguistic term, for both the input and output variables. These values were generally selected to balance the membership functions across the *universe of discourse*, or range of crisp values that a given variable can take. The only exception to this was that the membership function for the linguistic value *high* for the linguistic variable *distance to task* begins slightly earlier than it does relative to the other linguistic variables, to add a slight bias towards robots with a lower distance from the task. This was done to help better distribute tasks among robots. Regardless, the overlapping nature of the membership functions allows for smooth interpolation between the different input states.

2) Rule-base Design: The actual mapping of inputs to outputs within a fuzzy system is based on the rule-base of the system. For a system with n inputs, and m membership functions for every input, a fully defined rule-base consists of m^n rules. Therefore, the rule-base for this inference system consisted of 27 rules, as there were 3 inputs with 3 membership functions per input.

These rules represent every combination of membership values across all inputs. The rule-base was not designed with any specific preference between rules and instead was created based on the input level of the linguistic variables. The variable combinations and their corresponding output values are shown in Table I. As an example, if the load history of the robot is low, the distance to the current task is low, and the travel history of the robot is low, then the suitability of that robot for the given task is $very\ high$, as three variables reading low corresponds to a $very\ high$ suitability.

TABLE. I
RULE COMBINATIONS USED WITHIN THE FIS RULE-BASE.

Input Variable Set	**Output Variable**
$\{3L\}$	VH
$\{2L,\ 1M\}$	H
$\{2L,\ 1H\},\ \{1L,\ 2M\},\ \{1L,\ 1M,\ 1H\}$	M
$\{1L,\ 2H\},\ \{3M\},\ \{2M,\ 1H\}$	L
$\{1M,\ 2H\},\ \{3H\}$	VL

B. Implementing the FIS

To conceptualize how the FIS operates, it is best to give an example of its implementation and use. The FIS was designed using *scikit-fuzzy*, a fuzzy logic toolkit for Python that provides users with a robust array of fuzzy logic algorithms and allows for the creation and usage of fuzzy inference systems. This package allowed for the definition of the membership functions for each linguistic value, the instantiation of the rule-base, and the development and deployment of the FIS itself.

Development of the inference system first began by creating the input linguistic variables, which involved defining the linguistic variable itself, the universe of discourse for each of the variables, and the a, b, and c values for each membership function for each linguistic value. The universes of discourse were chosen to be $[0,\ 10]$, $[0,\ 25]$, and $[0,\ 50]$ for the load history, distance to task, and distance history respectively. As mentioned, the membership functions for low, $medium$, and $high$ were generally split evenly across these universes of discourse.

Following this, the output linguistic variable was created, which involved defining the linguistic variable, its universe of discourse, and the values of a, b, and c for each membership function for each linguistic value. The universe of discourse for suitability was selected to be $[0,\ 10]$, meaning that the value of suitability is a scalar metric that ranges from 0 to 10, with the most suitable robot being represented by a suitability of 10, and the least suitable robot therefore being represented by a suitability of 0. The five membership functions for suitability were split evenly across this universe of discourse.

The rule-base was created based on all m^n rule combinations and was defined using the relationship shown in Table I. Intrinsic to the *scikit-fuzzy* toolbox, a control system was then defined, which effectively creates a fuzzy inference class from the rule-base, which initializes the system and populates it with the fuzzy rules. Utilizing the FIS is simple, as it just needs to be provided the crisp values of each input linguistic variable and the crisp, defuzzified output is returned. The *scikit-fuzzy* toolbox handles the entirety of the fuzzification, rule evaluation, aggregation, and defuzzification, allowing for a simple, easy-to-use implementation of the FIS.

As an example of its usage, consider the following four robots, given within Table II. This scenario is representative of the one shown within Figure 1, where two classes of robots are bidding to head to the task location. Each robot has a capability (C), which can be either imagery (I) or measurement (M). The robots are also defined by their objective parameters, such as load history (LH), distance to the task (D), and distance history (DH). The parameters relating to distance are measured in meters. These values are then fed into the FIS, which amalgamates them into a scalar value of suitability (S). The bolded values of suitability represent the robots from each class that won this round of the auction.

TABLE. II
AN EXAMPLE OF A TASK ALLOCATION INSTANCE USING THE FIS.

Robot ID	**C**	**LH**	**D**	**DH**	**S**
1	**I**	2	14.024	34.047	4.11
2	**I**	0	35.213	0	**5.00**
3	**M**	1	15.374	18.783	5.53
4	**M**	1	14.620	11.881	**6.32**

As can be seen, Robot 1 has been to the task site twice and has travelled roughly 34 meters already. As such, its suitability value is lower than the other robots. Conversely, while Robot 2 is quite far from the task site, it has not yet been to the task site, and therefore receives a suitability value of 5.00, meaning that it is more suited than Robot 1 for this task, but overall has a medium-suitability value. If there were a robot that was closer to the task than Robot 2 while still having not been to the task site, it would receive a higher suitability and would therefore win for the imagery robots. Similarly, for the measurement robots, Robot 4 is closer to the task and has travelled less thus far, and therefore has a higher suitability for the task at hand than that of Robot 3. The inference time for the system to infer the suitability of all four robots was 0.315 seconds.

This value of suitability would then be placed within the bid structure shown in Equation 1, and sent forth to the auctioneer, who would select the robot with the highest suitability and matching capability to that of the task requirement. This simple but important scenario shows how the objective parameters of a robot can be used to infer how suitable a given robot is for the task at hand in a manner that is easy to understand and intuitive. Together with the capability of a robot, the value of suitability can be used to describe both what a robot is capable of doing and how well poised it is to achieve the task.

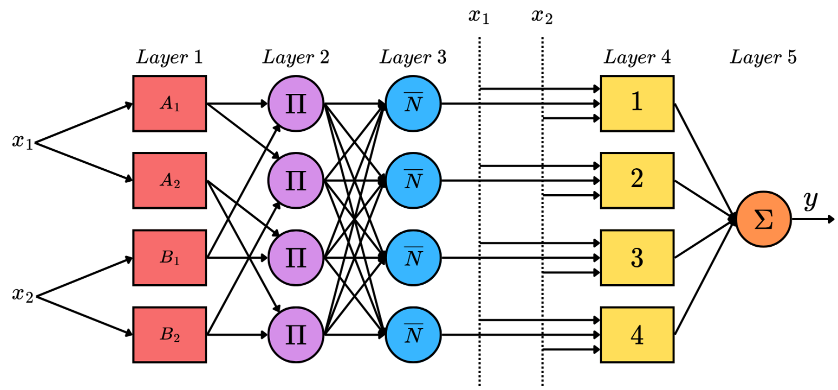

Figure. 2. Architecture of an ANFIS with two inputs, two membership functions per input, and four rules.

IV. ANFIS DESIGN

While the current methodology employs the use of a FIS, the focus of the research then turned towards the use of machine learning to perform regression between objective parameters and the suitability of a given robot. The representation of a FIS using the structure of a neural network offers benefits in the form of parallelized computation and the ability to learn and extract a higher dimensional relationship between the input variables, providing a further mapping between the objective parameters and the suitability of a given robot, all while leveraging parallelization to decrease inference time.

Figure 2 shows the basic architecture of an ANFIS with two inputs, four membership functions (two per input), and four rules. As can be seen, there are five key layers to this architecture. The square nodes are $adaptive\ nodes$, in that they have their values updated through training the network, whereas the circular nodes are $fixed$. Let the output of each node in each respective layer be given by O_k^j, where k is the k^{th} node of layer j.

The first layer represents the membership layer, where the crisp input values are fuzzified using Equation 3. This layer effectively determines the degree of membership of an input to a membership function. x_1 and x_2 represent the crisp values of the input variables, and $\mu_{A_i}(x_1)$ and $\mu_{B_i}(x_2)$ denote the membership functions of the linguistic values A_i and B_i, respectively, for $i = 1, 2$. The parameters of this layer are called the *premise parameters* and represent the defining variables of the membership functions. For triangular membership functions, these would be the a, b, and c parameters.

$$O_k^1 = \mu_{A_i}(x_1) \quad ; \quad O_k^1 = \mu_{B_i}(x_2) \tag{3}$$

The second layer calculates the accumulated firing strength of the rule antecedents, meaning that it calculates the degree to which a rule fires based on the membership values of the inputs. This is achieved using a *t-norm* operator, such as in Equation 4, where ω_k is the firing strength.

$$O_k^2 = \omega_k = \mu_{A_i}(x_1) \times \mu_{B_i}(x_2) \tag{4}$$

The third layer calculates the normalized firing strength, $\bar{\omega}_k$, using Equation 5.

$$O_k^3 = \bar{\omega}_k = \frac{\omega_k}{\sum_k \omega_k} \tag{5}$$

The fourth layer calculates the output of each rule based on a linear function of the inputs, which is then weighted by the normalized firing strength of that rule. This is done using Equation 6. Linear systems are used to describe the local dynamics of each rule. The parameters in this layer are called *consequent parameters*, and are given by p_k, q_k, and r_k.

$$O_k^4 = \bar{\omega}_k f_k = \bar{\omega}_k(p_k x_1 + q_k x_2 + r_k) \tag{6}$$

Finally, the output of the network sums the linear combinations of each consequent output, and is given by Equation 7.

$$O_k^5 = \sum_k \bar{\omega}_k f_k \tag{7}$$

Each layer was designed and implemented using Keras, the high-level API for TensorFlow within Python. Custom layers were defined by subclassing the default Keras layer, ensuring that the developed model could be implemented using the Keras functional API. The FIS that had been designed previously was utilized as a data generator by randomly sampling between the universe of discourse of each linguistic variable and using the FIS to infer suitability from these values. This was done to generate a dataset of task allocation mappings, which was then used to train the ANFIS.

The premise and antecedent parameters were updated in training using gradient descent, which was handled using TensorFlow's automatic differentiation algorithm. Due to the custom nature of the ANFIS implementation, it was designed such that it could handle any number of inputs, membership functions per input, and type of membership function, provided that they were continuous and piecewise differentiable. After performing a hyperparameter sweep, the model that approximated the FIS the best had a structure defined by five generalized bell membership functions per input, trained with a learning rate of 0.0005 for 500 epochs using a batch size of 128.

This model was then trained and saved using Keras' implicit support for saving and redeploying models, which was done so that the model could be deployed and tested. The model was then tested in a similar task allocation scenario as done with the FIS in Table II. It was found that the ANFIS model was able to perform the inference for all four robots in 0.032 seconds, representing a reduction in inference time by a factor of almost ten. The model did not produce the exact same output of suitability, but still preserved the relationship between the objective parameters and the suitability of each robot. The outcome of the auction was the same, with Robot 2 being selected for the imagery robots, and Robot 4 being selected for the measurement robots, as shown in Table III.

TABLE. III
ANFIS TASK ALLOCATION RESULTS.

Robot ID	C	LH	D	DH	S
1	**I**	2	14.024	34.047	4.33
2	**I**	0	35.213	0	**4.79**
3	**M**	1	15.374	18.783	5.36
4	**M**	1	14.620	11.881	**5.91**

V. CONCLUSIONS & FUTURE WORK

Within this paper, a unique methodology was presented for formulating bids within auction-based task allocation for the MRTA problem. A fuzzy inference system was used to infer the suitability of a given robot based on its load history, distance to the current task, and the distance that they have travelled thus far. This scalar value of suitability is then combined with the capability of the robot and put forth in a bid. By utilizing an FIS to intuitively reason how suitable a robot is for a given task, complex combinatorial optimization solutions are avoided. To augment this approach, an adaptive neuro-fuzzy inference system was developed using a custom implementation within Keras, and trained on the FIS data. This resulted in comparable inference in terms of the ANFIS' ability to replicate the FIS results, but the ANFIS was able to do this 9.8x faster, achieving a blindingly fast inference time of 0.032 seconds. This ready-to-use model can now be deployed on robotic agents for inference.

However, this work is not without its limitations. This methodology requires a robust communication framework to ensure bidirectional communication between the auctioneer and the participants of the auction, as is the case with all auction-based solutions to the MRTA problem. Moreover, when the objective parameters of a robot exceed their universes of discourse, the suitability calculation will suffer. A proposed method of alleviating this is to have robots return to a charging or rest station to have their parameters reset when they approach a certain threshold, to prevent their objective parameters from ever exceeding their universes of discourse. Future work would involve exploring the use of a traditional artificial neural network trained on the FIS data. This would be done to keep the parallelization and fast inference time but hopefully extract a deeper relationship between objective parameters, such that predictions better reflect the performance of the FIS. Furthermore, this methodology needs to be evaluated first through simulation, and then through experimentation on real robotic agents using ROS2. High-fidelity simulations are to be conducted using a physics engine such as Gazebo or Unity3D. Finally, this methodology needs to be compared against existing methods of task allocation to validate its applicability and efficiency.

REFERENCES

[1] L. Leong, J. Seau, K. Teo, K. Tze, and Y. H. Pin, "Four wheeled mobile robots: A review," in *2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*, 2022, pp. 1–6. [Online]. Available: https://doi.org/10.1109/IICAIET55139.2022.9936855

[2] S. N., K. C. R.M., R. M.M., and M. N. Janardhanan, "Review on state-of-the-art dynamic task allocation strategies for multiple-robot systems," *Industrial Robot: the international journal of robotics research and application*, vol. 47, no. 6, pp. 929–942, Sep. 2020. [Online]. Available: https://www.emerald.com/insight/content/doi/10.1108/IR-04-2020-0073/full/html

[3] K. F. E. Tsang, Y. Ni, C. F. R. Wong, and L. Shi, "A novel warehouse multi-robot automation system with semi-complete and computationally efficient path planning and adaptive genetic task allocation algorithms," in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2018, pp. 1671–1676. [Online]. Available: https://doi.org/10.1109/ICARCV.2018.8581092

[4] G. Rishwaraj and S. Ponnambalam, "Integrated trust based control system for multirobot systems: Development and experimentation in real environment," *Expert Systems with Applications*, vol. 86, pp. 177–189, 2017. [Online]. Available: https://doi.org/10.1016/j.eswa.2017.05.074

[5] A. Khamis, A. Hussein, and A. Elmogy, "Multi-robot task allocation: A review of the state-of-the-art," *Cooperative Robots and Sensor Networks 2015*, pp. 31–52, 2015. [Online]. Available: https://doi.org/10.1007/978-3-319-18299-5_2

[6] M. Alshaboti, "Multi-robot Task Allocation System: Fuzzy Auction-Based and Adaptive Multi-threshold Approaches," *SN Computer Science*, vol. 2, no. 81, 2021. [Online]. Available: https://doi.org/10.1007/s42979-021-00479-x

[7] G. Heppner, D. Oberacker, A. Roennau, and R. Dillmann, "Behavior tree capabilities for dynamic multi-robot task allocation with heterogeneous robot teams," *Proceedings of the IEEE International Conference on Robotics and Automation 2024*, pp. 1–8, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2402.02833

[8] R. K, B. R, P. Panchu K, and R. M, "A novel fuzzy and reverse auction-based algorithm for task allocation with optimal path cost in multi-robot systems," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 5, p. e6716, 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.6716

[9] A. Zenati, N. Aouf, and O. Kechagias-Stamatis, "Distributed Fuzzy Semi-Infinite Auction Based Optimization for Cooperative Robots Tasks Allocation," in *2022 American Control Conference (ACC)*, Jun. 2022, pp. 1898–1903. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9867692

[10] O. Valero, J. Antich, A. Tauler-Rosselló, J. Guerrero, J.-J. Miñana, and A. Ortiz, "Multi-robot task allocation methods: A fuzzy optimization approach," *Information Sciences*, vol. 648, p. 119508, Nov. 2023. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0020025523010939

[11] Z. Ma and H. Gong, "Heterogeneous multi-agent task allocation based on graph neural network ant colony optimization algorithms," *Intelligence & Robotics*, vol. 3, no. 4, pp. 581–595, Oct. 2023. [Online]. Available: https://doi.org/10.20517/ir.2023.33

[12] B. P. Gerkey and M. J. Matarić, "A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems," *The International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, Sep. 2004. [Online]. Available: https://doi.org/10.1177/0278364904045564