Scalable PSO-Enhanced Fuzzy Decision-Making for Decentralized Task Allocation in Multi-Robot Systems

Hamza Teggar [*,**] **Kheireddine Mekkaoui** [***]
Abdessamed Ouessai [*,****] **Amina Houari** [*,†]
Mohammed Salem [*,‡]

[*] *LISYS Lab, Mustapha Stambouli University of Mascara, Algeria.*
[**] *(e-mail: hamza.teggar@univ-mascara.dz).*
[***] *Department of computer sciences, University of Saida - Dr Moulay Tahar, Algeria(e-mail: Kheireddine.mekkaoui@univ-saida.dz)*
[****] *(e-mail: ouessai@univ-mascara.dz).*
[†] *(e-mail: amina.houari@univ-mascara.dz).*
[‡] *(e-mail: salem@univ-mascara.dz).*

Abstract: This paper presents a novel fuzzy decision-making framework for decentralized task allocation in heterogeneous multi-robot systems, enhanced through particle swarm optimization. The proposed model, called Particle Swarm Optimization of Fuzzy Decision-Making for Task Selection (PSO-FDM-TS), automates the tuning of fuzzy membership function parameters to improve adaptability and robustness in uncertain environments with limited communication. Through extensive simulations in a warehouse transportation scenario, PSO-FDM-TS achieves a significant reduction in task completion error compared to both manually-tuned fuzzy systems and observation-based allocation approaches. Moreover, the results show that PSO-FDM-TS maintains superior performance across diverse operational conditions, including varying task heterogeneity, different robot team sizes, and large-scale deployments without the need for inter-robot communication. This work represents a key step toward scalable and efficient coordination of heterogeneous robot teams in real-world applications such as automated logistics and mission adaptation in hostile environments.

Keywords: Multi-robot systems; task allocation; fuzzy logic; particle swarm optimization; heterogeneous robots; decentralized decision-making.

1. INTRODUCTION

Heterogeneous Multi-Robot Systems (HMRS) have recently emerged as a powerful and flexible paradigm in robotics. Unlike homogeneous teams, HMRS are composed of autonomous robots that differ in sensing, actuation, and computational capabilities. This diversity enables more efficient collaboration in missions that would otherwise be difficult for homogeneous teams or single robots (Verma et al., 2025; Do et al., 2025). The heterogeneity enables specialization of roles, where each robot can be assigned tasks aligned with its unique strengths, improving overall mission efficiency, robustness to failures, and adaptability to dynamic environments (Mayya et al., 2021).

One prominent and practical application domain of HMRS is automated warehouse management. Modern warehouses rely heavily on fleets of autonomous robots to perform various tasks such as inventory transport, shelf replenishment, order picking, and sorting. In such environments, robots may differ widely; for example, some are specialized in heavy lifting, others in fast navigation, and others in fine manipulation. Efficient coordination of these heterogeneous robots is essential to maintain high throughput, reduce costs, and adapt to changing order demands and layouts (Oliveira et al., 2022). The complexity of warehouse environments, combined with noisy sensing and unreliable communication, highlights the need for robust decentralized task-allocation strategies that can handle uncertainty in real time.

To address the robot's uncertainty and limited observability when making decisions to perform a given task, fuzzy logic (Zadeh, 1965) has emerged as a robust and intuitive framework. By representing system variables with linguistic terms and fuzzy membership functions, fuzzy logic captures the inherent imprecision of sensor data and environmental factors, enabling flexible, human-like reasoning in multi-agent systems (Rafik et al., 2022; Thi et al., 2022). In the context of multi-robot task allocation, fuzzy logic can model complex decision criteria, incorporate expert knowledge and tolerate ambiguity, thereby improving coordination under uncertainty (Mulani and Ramachandran, 2023; Xidias and Zacharia, 2024).

Building on these insights, previous studies (Rechache et al., 2024; Teggar et al., 2021b,a) introduced the Fuzzy Decision-Making for Task Selection (FDM-TS) framework, a decentralized and communication-free approach wherein each robot employs fuzzy inference to autonomously allocate tasks based solely on locally observed information. The framework demonstrated promising results by handling uncertainty effectively while achieving decentralized task allocation without requiring significant communication. However, several limitations still hinder its practical deployment. One of the most critical limitations arises from the dependence on manual calibration of fuzzy membership functions, a procedure that is not only time-consuming but also demands substantial domain-specific expertise. Manual tuning typically produces static parameter settings that lack the flexibility needed to adapt to evolving mission objectives or changing environmental conditions. In addition, as the scale of the system increases with a growing number of robots and tasks, the dimensionality of the parameter space expands combinatorially, rendering exhaustive manual optimization computationally infeasible and practically unmanageable.

To overcome these limitations, this study proposes integrating automated optimization techniques, with a particular focus on Particle Swarm Optimization (PSO), a metaheuristic well suited for tuning fuzzy systems. PSO simulates the collective behavior of swarms to efficiently search complex, high-dimensional, and nonlinear parameter spaces, making it an attractive choice for optimizing fuzzy membership functions (Aliskan, 2025). Its population-based search strategy balances exploration and exploitation, facilitating convergence to near-optimal solutions without requiring gradient information. Additionally, PSO's relative simplicity and computational efficiency enable practical implementation in real-time or near-real-time multi-robot applications.

The enhanced Particle Swarm Optimization for Fuzzy Decision-Making in Task Selection (PSO-FDM-TS) framework presented in this work provides a substantial contribution by automating membership function tuning, thereby reducing manual effort while improving system adaptability and scalability. Through extensive experimental evaluation in simulated environments that incorporate realistic communication constraints and sensor noise, the proposed approach demonstrates marked improvements in task allocation accuracy and reductions in mission completion errors. Importantly, these benefits are achieved without sacrificing the decentralized architecture or scalability inherent to the original framework.

In summary, this work contributes to the growing body of research seeking robust, scalable, and adaptive coordination mechanisms for heterogeneous multi-robot systems operating under uncertainty. By combining the interpretability of fuzzy logic with the optimization capability of PSO, this approach offers a novel solution that addresses critical challenges in decentralized task allocation, paving the way for more effective deployment of HMRS in complex real-world missions, such as automated warehouse management, search and rescue operations, and environmental monitoring.

The remainder of this paper is organized as follows: Section 2 analyzes existing works on task allocation and fuzzy optimization. Section 3 formally models the task allocation problem in heterogeneous systems. Section 4 presents the FDM-TS mechanism, whose limitations motivate the integration of PSO optimization described in Section 5. Sections 6 and 7 experimentally validate the approach and analyze the performance. Finally, a conclusion summarizes the contributions and future perspectives.

2. RELATED WORKS

Task allocation in heterogeneous multi-robot systems has been widely studied. Existing approaches are commonly classified into three categories: centralized, market-based, and fully distributed strategies (Gerkey and Mataric, 2004; Rizk et al., 2019). Centralized methods exploit global system information to optimize task assignments but suffer from scalability limitations and single points of failure in large-scale deployments (Gerkey and Mataric, 2004). Market-based approaches, such as the contract-net protocol, decentralize decision-making by allowing robots to bid on tasks (Zhao and Wang, 2013). Although robust, these methods often require substantial communication and remain sensitive to miscoordination. Fully distributed strategies rely on local decision rules and utilities derived from individual robot perceptions (Mayya et al., 2021). These approaches are generally robust to failures and adaptable to dynamic environments, but they require well-designed coordination rules to avoid deadlocks and conflicting task assignments.

To improve efficiency, several task-decomposition strategies have been proposed. These methods divide complex missions into smaller, often independent subtasks that can be executed in parallel, improving the use of heterogeneous robot capabilities (Rizk et al., 2019; Ching-Wei and Wei-Yu, 2024). Task decomposition can be performed manually (Gil et al., 2023) or automatically. In both cases, it allows robots to make local task-selection decisions without requiring explicit communication. For instance, Gerkey and Mataric (2004) introduced a mathematical framework for dynamic task allocation under communication constraints using local utility functions. Similarly, Dahl et al. (2009) proposed a vacancy-chain approach for distributed task allocation, optimizing system-wide objectives such as efficiency and cost-effectiveness.

Recent research has focused on resilient task allocation for heterogeneous teams, often relying on AI-driven techniques to enhance coordination, adaptability, and decision-making under uncertainty. Notable approaches include fuzzy logic–based models for task allocation (Teggar et al., 2021b,a; Valero et al., 2023; Jaume-Martin et al., 2024; Rechache et al., 2024). In Multi-Robot Task Allocation System (MRTAS), fuzzy logic has been employed through Fuzzy Inference Systems (FIS) to combine multiple decision criteria into utility-based decision-making. These criteria commonly include factors such as distance, energy availability, and task urgency (Valero et al., 2023). Most applications focus on energy-aware task execution in bio-inspired simulations (Jaume-Martin et al., 2024; Islam and Akhtar, 2017; Valero et al., 2023). Others investigate parallel dynamic allocation using the Fuzzy Predictor for Task Execution (FP-TE) (Teggar et al., 2021b,a).The FP-TE output delivers a utility value that serves as the allocation criterion, enabling each robot to autonomously determine its next task. However, the majority of these approaches remain confined to simulated or communication-rich environments, which restricts their scalability and applicability in realistic decentralized scenarios. Addressing this limitation, Rechache et al. (2024) proposed a Fuzzy Decision-Making in Task Selection (FDM-TS) for HMRS without explicit inter-robot communication, where complex missions are decomposed into elementary subtasks managed by a FP-TE. Despite such advances, important challenges persist regarding scalability, robustness to uncertainty, and real-world deployment. Moreover, most fuzzy-based MR-TAS still rely on expert-defined membership functions and rule weights (Valero et al., 2023). This limits their adaptability to heterogeneous missions, dynamic environments, and diverse robotic platforms.

To address these limitations, several studies have explored automatic optimization of fuzzy systems using metaheuristics such as Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) (Roy et al., 2022). PSO is particularly attractive due to its simplicity, low computational cost, and fast convergence in high-dimensional continuous spaces (Kennedy and Eberhart, 1995). It has been successfully applied to optimize fuzzy controllers in navigation, industrial process control, and energy management tasks (Prauzek et al., 2025). Nevertheless, the integration of fuzzy logic and PSO for fully decentralized task allocation in HMRS remains largely unexplored.

The review of existing approaches shows that most fuzzy-based task allocation methods rely on fixed, expert-defined membership functions, which limits their adaptability in heterogeneous or dynamic environments. Similarly, optimization-based strategies typically assume some form of communication or negotiation between robots, making them unsuitable for strictly communication-free settings. Furthermore, prior studies rarely address the inherent combinatorial nature of decentralized task allocation, which leads to NP-complete complexity when considering multiple robots and heterogeneous subtasks. These limitations motivate the development of a more adaptive and scalable framework.

The originality of this work, detailed in Sections 4 and 5, lies precisely in addressing these gaps by combining decentralized fuzzy reasoning with PSO-based optimization of membership functions, all operating under strict communication constraints. Accordingly, Section 3 is dedicated solely to formalizing the underlying problem and its inherent complexity, while the proposed methodology is introduced immediately afterward in the subsequent sections.

3. PROBLEM FORMULATION

A team of g autonomous robots $R = r_1, r_2, \ldots, r_g$ is considered, operating in a structured environment without a central controller and without direct communication between robots.

The global mission consists of executing a complex task T, observed only at the beginning of the mission in a specific observation area. This initial observation provides each robot with a full description of the elementary tasks, which cannot be accessed again later in the mission. Consequently, robots must store this information and make their future decisions solely based on this observation and their local perceptions.

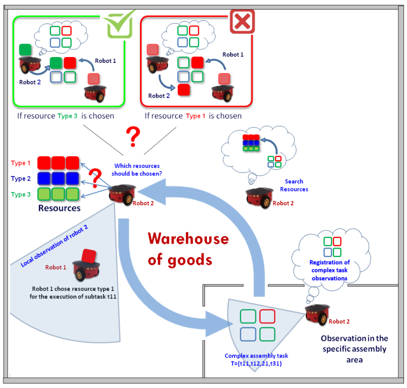


Fig. 1. Task Allocation Process Based on Local and Global Observations in a Multi-Robot System

Fig. 1 illustrates the resource selection process in a modern warehouse managed by mobile robots. In this scenario, *Robot2* first observes the description of a complex task in the assembly area and then searches for the required resources within the warehouse. Once the resources are identified, the robot initiates a decision-making process to determine the most suitable choice. This decision relies on two key elements: (i) the memorized observation of the complex task to be executed, and (ii) the local observation of resources already selected by other robots. For example, *Robot1* has already chosen a *Type1* resource for the execution of subtask t_{11}. Since the assembly requires only one unit of this type, *Robot2* must instead select an alternative resource, either *Type2* or *Type3*, thereby ensuring complementarity and efficiency in the collective task allocation process.

3.1 Decomposition of complex task

The complex task T consists of n subtask (or n simple tasks):

$$T = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}, \tag{1}$$

where each subtask σ_i of type i is composed of $|\sigma_i|$ elementary tasks:

$$\sigma_i = \{t_{i1}, t_{i2}, \ldots, t_{i|\sigma_i|}\}. \tag{2}$$

An elementary task $t_{ij} \in \sigma_i$, with $1 \leq i \leq |T|$ and $1 \leq j \leq |\sigma_i|$, is defined as a minimal action that cannot be further decomposed. Such tasks are mutually independent, without precedence constraints, and can therefore be executed in parallel by different robots.

The total number of elementary tasks in the mission is therefore:

$$N = \sum_{i=1}^{|T|} |\sigma_i|. \tag{3}$$

3.2 Resources and Environment

Each elementary task t_{ij} requires a specific quantity of physical resources (e.g. goods), which are distributed across a set of nodes V_{res} located outside the observation area. The available stock of a resource type ℓ at node v is denoted $b_v^\ell \in \mathbb{N}$. The environment is represented as a metric graph $G = (V_{res}, E)$, where $E \subseteq V \times V$ the set of edges connecting them. A travel cost function $d : V_{res} \times$

$V_{res} \to \mathbb{R}^+$ is associated with each edge, defining the cost of moving between two nodes.

***Remark** 3.1.* A robot can execute an elementary task t_{ij} only if it has previously acquired the required resources.

3.3 Desired Ratios and Progress State

In Fig. 2, the **desired ratio** $\tau(\sigma_i)$ of subtask σ_i denotes the intended proportion of elementary tasks composing σ_i relative to the overall complex task:

$$\tau(\sigma_i) = \frac{|\sigma_i|}{N}. \tag{4}$$

Similarly, the **progress state** $P(\sigma_i)$ of subtask σ_i is defined as the proportion of its executed elementary tasks with respect to the total number of tasks:

$$P(\sigma_i) = \frac{|\sigma_i'|}{N}, \tag{5}$$

where σ_i' denotes the set of elementary tasks of σ_i that have already been completed by the robots. Each completed elementary task is added to the corresponding set of the same type i under the index t_{ix}, with:

$$x = |\sigma_i'| + 1 \tag{6}$$

3.4 Global Error and Global Accuracy

The **relative error** in the execution of a subtask σ_i is defined as the difference between its current progress and its desired completion ratio:

$$\delta(\sigma_i) = P(\sigma_i) - \tau(\sigma_i). \tag{7}$$

***Remark** 3.2.* Initially, the relative error $\delta(\sigma_i)$ is typically negative because the progress $P(\sigma_i)$ starts from zero while the desired ratio $\tau(\sigma_i)$ is strictly positive. As execution progresses, $\delta(\sigma_i)$ increases and tends to converge towards zero if the allocation strategy is effective.

The **global error** $\Delta(T)$ corresponds to the sum of absolute relative errors over all subtasks:

$$\Delta(T) = \sum_{i=1}^{|T|} |\delta(\sigma_i)|. \tag{8}$$

Finally, the **accuracy** of the mission $A(T)$ is defined as the inverse of the global error when the error is strictly positive, and is set to a constant value $(1/\varepsilon)$ when the global error is equal to zero or negligible error $\varepsilon \neq 0$:

$$A(T) = \begin{cases} \dfrac{1}{\Delta(T)} & if \Delta(T) > \varepsilon \\ 1/\varepsilon & if \Delta(T) \leq \varepsilon \end{cases} \tag{9}$$

where ε it may also serve as a stopping criterion for the allocation process.

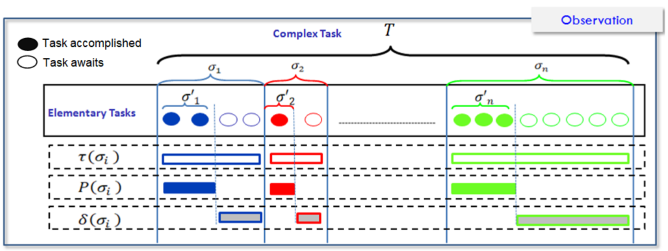


Fig. 2. Evaluation of the complex task execution.

3.5 Optimization Objective

The objective of MRTAS is to execute the elementary tasks so that the final distribution of completed subtasks matches the desired ratios $\tau(\sigma_i)$ as closely as possible, while accounting for resource availability and local feasibility constraints.

Formally, the problem can be stated as:

$$\min_{\pi} \ \Delta(T) = \sum_{i=1}^{|T|} |\delta(\sigma_i)|, \tag{10}$$

where π denotes the strategies for the selection of elementary task by all robots over time.

***Remark** 3.3.* Minimizing the global error $\Delta(T)$ is equivalent to maximizing the accuracy $A(T)$:

$$\max_{\pi} \ A(T) \quad \Leftrightarrow \quad \min_{\pi} \ \Delta(T). \tag{11}$$

The optimization is subject to the following constraints:

Progress constraint: A robot r may select a task belonging to σ_i only if the current progress $P(\sigma_i)$ is below its desired ratio $\tau(\sigma_i)$:

$$P(\sigma_i) < \tau(\sigma_i), \quad \forall i \in \{1, \dots, |\sigma_i|\}. \tag{12}$$

Resource constraint: Let $a_{i\ell}$ denote the amount of resource type ℓ required to execute an elementary task $t_{ij} \in \sigma_i$, and $b_{v\ell}$ the available stock at resource node v. A robot r can execute the elementary task t_{ij} only if:

$$\sum_{v \in V_{\text{res}}} u_{rv\ell} \geq a_{i\ell}, \quad \forall \ell, \tag{13}$$

where $u_{rv\ell}$ is the quantity of resource ℓ accessible to r from node v.

Feasibility constraint: Let $\mathcal{F}(r, \sigma_i)$ be the set of feasible actions for robot r given its current location, resources, and environment state. A selected task must belong to the feasible set:

$$\sigma_i \in \mathcal{F}(r, \sigma_i). \tag{14}$$

The overall problem can thus be summarized as:

$$\begin{aligned} &\min_{\pi} \sum_{i=1}^{|T|} |P(\sigma_i) - \tau(\sigma_i)|, \\ &\text{subject to:} \\ &\quad P(\sigma_i) < \tau(\sigma_i), \quad \forall i, \\ &\quad \sum_{v \in V_{\text{res}}} u_{rv\ell} \geq a_{i\ell}, \quad \forall \ell, \\ &\quad \sigma_i \in \mathcal{F}(r, \sigma_i). \end{aligned} \tag{15}$$

This formulation makes explicit the trade-off between minimizing the task allocation error and respecting operational constraints in a decentralized multi-robot environment.

3.6 Task Allocation Complexity and Strategy Selection

Building upon the constraints imposed by the selection of elementary tasks, it is important to emphasize that the subsequent allocation process inherently involves a combinatorial search. In a distributed multi-robot system (DMRS), this search can be formalized as a combinatorial optimization problem, where the objective is to identify the optimal matching between available robots and feasible subtasks. More specifically, the problem can be expressed as the selection of a subset of *(robot, task)* pairs from the set of all admissible arrangements, without repetition. The total number of possible assignments, denoted by NB_{Assgn}, is expressed as:

$$NB_{\text{Assgn}} = \frac{NB_{\text{pt}}!}{(NB_{\text{pt}} - |R|)!} \quad \text{when} \quad NB_{\text{pt}} > |R| \tag{16}$$

Here, $|R|$ is the number of robots in the system, and NB_{pt} represents the number of all *elementary tasks* $t_{ij} \in \sigma_i$ whose progress rate $P(\sigma_i)$ has not yet reached the desired completion ratio $\tau(\sigma_i)$. Formally:

$$NB_{\text{pt}} = \sum_{\substack{i=1 \\ P(\sigma_i)<\tau(\sigma_i)}}^{|T|} (|\sigma_i| - |\sigma_i'|) \tag{17}$$

where $|\sigma_i|$ denotes the total number of subtasks in σ_i, and $|\sigma_i'|$ the number of already completed subtasks as shown in (5).

This class of task-allocation problems is known to be NP-complete, meaning that computation time grows exponentially with problem size and makes exact solutions impractical for large-scale systems.

4. FUZZY DECISION-MAKING IN TASK SELECTION (FDM-TS)

To derive an effective allocation strategy π, as defined in (15), this section introduces a task allocation model for multi-robot systems that integrates a fuzzy predictor. The predictor leverages the values obtained from (18) to compute a utility function that simultaneously accounts for task requirements and contextual factors. This utility function provides the foundation for decision-making in resource selection and task distribution. By embedding fuzzy reasoning, the model is capable of addressing uncertainty and imprecision inherent in real-world environments, thereby improving the robustness of the allocation mechanism.

Before detailing the formal specification of the utility function and its implementation within the proposed framework, it is worth recalling that the robot r can only choose an elementary task $t_{ij} \in \sigma_i$ only if the two conditions specified in (12) and (13) are satisfied. If these conditions hold, and under the assumption that when a robot considers executing an elementary task $t_{kj} \in \sigma_k$, the predicted error of σ_k is given by:

$$\delta'(\sigma_k) = \frac{|\sigma_k'| + 1}{N} - \tau(\sigma_k), \tag{18}$$

while the corresponding predicted global error resulting from this choice is given by:

$$\Delta'(T, k) = \left|\delta'(\sigma_k)\right| + \sum_{i \neq k}^{|T|} \left|\delta(\sigma_k)\right|. \tag{19}$$

In a single-robot scenario, a simple strategy based on selecting the task that minimizes the predicted future error may be effective , but it often leads to coordination challenges in multi-robot environments. Specifically, multiple robots may select the same elementary task t_{kj} simultaneously, resulting in overshooting its target execution ratio and lower overall assignment accuracy of σ_k.

To address this issue, a fuzzy predictor inspired by FP-TE (Teggar et al., 2021b) is employed. The predictor operates locally, without inter-robot communication, and uses the concept of fuzzy logic to calculate a task utility value that will be included in the strategy of task selection. This utility considers both the potential to reduce the global complex task error and the likelihood that the task will be chosen by other robots, thus improving the distribution of work and maintaining cooperative behavior in the absence of explicit coordination.

Predictor Inputs: For each candidate future elementary task t_{kj}, the FP-TE uses two inputs:

- *Predicted complex task error* $PCE(T, k)$: the error value expected if t_{kj} is executed next, computed as:

$$PCE(T, k) = \delta'(\sigma_k) + \sum_{i \neq k}^{|T|} \delta(\sigma_i) \tag{20}$$

  where $\delta'(\sigma_k)$ is the predicted relative error of simple task σ_k if t_{kj} is chosen, calculated using (18). It is worth noting that $PCE(T, k)$ takes positive or negative values, unlike $\Delta'(T, k)$, which takes positive values as given in (19).
- *Local observation of active robots* $\mathrm{D}(r)$: the estimated density of active robots, measured as the number of robots detected in the vicinity using onboard perception (e.g. vision, sonar).

Predictor outputs: the output is the utility value $u(t_{fj})$ such that the utility function is defined as :

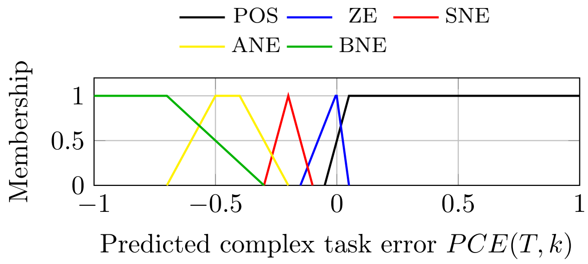
$$u : \sigma_f \longrightarrow [0, 1], \qquad u(t_{fj}) \in [0, 1], \;\; t_{fj} \in \sigma_f. \tag{21}$$

where for each elementary task $t_{fj} \in \sigma_f$, the utility function assigns a value between 0 and 1 such that the highest values indicate the most important tasks and which contribute more effectively to minimizing the global error $\Delta(T)$.

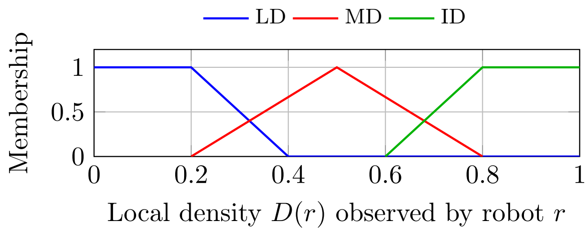
4.1 Fuzzy Inference System

Fig. 3 illustrates the FP-TE inputs and output, each represented as a linguistic variable defined by its corresponding membership functions:

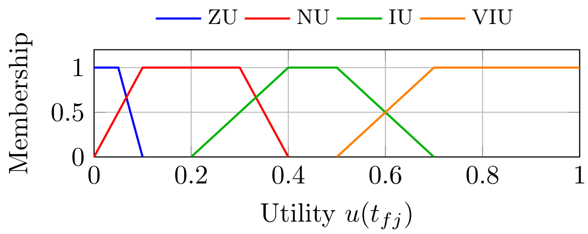
- *Predicted complex task error (Fig. 3a)*: ZE (Zero Error), SNE (Small Negative Error), ANE (Average Negative Error), BNE (Big Negative Error), POS (Positive)
- *Local observation (Fig. 3b)*: LD (Low Density), MD (Moderate Density), ID (Intensive Density)
- *Task utility (Fig. 3c)*: ZU (Zero Utility), NU (Negligible Utility), IU (Important Utility), VIU (Very Important Utility)



(a) Predicted error



(b) Local observation



(c) Predicted utility

Fig. 3. Fuzzy decision system with membership functions for predicted error, Local observation of active robots, and predicted utility.

The proposed FDM-TS employs the fuzzy inference system originally introduced by Mamdani and Assilian (1975). For a given elementary task t_{fj}, the corresponding linguistic value of $u(t_{fj})$ is derived based on the set of fifteen fuzzy rules summarized in Table 1, which will be used in the same way as the following example:

IF $PCE(T, k)$ is ZE AND $D(r)$ is LD THEN $u(t_{fj})$ is VIU

Table 1. Fuzzy rules for task utility evaluation

Expected error $PCE(T, k)$	**LD**	**MD**	**ID**
POS (Positive)	ZU	ZU	ZU
ZE (Zero Error)	VIU	NU	ZU
SNE (Small Neg. Error)	IU	NU	NU
ANE (Avg. Neg. Error)	NU	IU	IU
BNE (Big Neg. Error)	ZU	NU	IU

The rules encode the principle that if the robot density observed $D(r)$is high and the predicted error $PCE(T, k)$ of elementary task t_{fj} is small, the utility of t_{fj} that should be lowered to avoid multiple robots choosing it. Conversely, tasks with slightly higher predicted errors but lower competition potential should be given higher utility.

Finally, the crisp utility value noted by $u^*_{t_f}$ is obtained by applying defuzzification using the center of gravity method given by:

$$u^*_{t_f} = \frac{\sum_{i=1}^{n} y_i \cdot \mu_U(y_i)}{\sum_{i=1}^{n} \mu_U(y_i)} \tag{22}$$

where n is the number of the linguistic label (in this work $n = 4$). Each linguistic label is mapped to a numerical center value y_i, and its activation degree $\mu_U(y_i)$ is obtained after fuzzy inference.

4.2 FDM-TS Algorithm

Algorithm 1 enables fully decentralized task allocation in multi-robot systems without explicit inter-robot communication. Each robot independently determines its next elementary task t_{fj} by evaluating all available subtasks σ_i and it calculates the predicted error $PCE(T, k)$ that will result if t_{fj} is executed. In parallel, the robot estimates the local active robot density $D(r_k)$ using onboard perception. These two inputs are processed by fuzzy inference system of FP-TE with a fixed rule base to produce a crisp utility value $u^*_{t_k}$ according to (22) for each candidate task.

Algorithm 1 FDM-TS Algorithm

Input: Complex task T, initial observation $\mathcal{C}$, resources $b_{v\ell}$, membership functions, fuzzy rule table
Output: Elementary Task t_{fj} to execute
Mark mission not completed as *true*
while mission not completed **do**
  for $k \leftarrow 1$ to $|T|$ **do**
    if $P(\sigma_k) \geq \tau(\sigma_k)$ **or** insufficient resources **then**
      Mark k as *inadmissible*
      continue
    end if
    Compute $PCE(T, k)$ according to (20)
    Measure $D(r)$
    Fuzzify the inputs and apply fuzzy rules
    Defuzzify to obtain $u^*_{t_k}$ according to (22)
  end for
  if $k < |T|$ **then**
    Select t_{fj} with maximum $u^*_{t_k}$
    $j \longleftarrow |\sigma'_f| + 1$ according to (6)
    Retrieve required resources and execute t_{fj}
    Update σ'_f
    Update $P(\sigma_f)$ according to (5)
  else
    Mark mission not completed as *false*
  end if
end while

Once the elementary task is chosen under the first index f belonging to the subtask σ_f, the second index j will be updated according to (6). Finally the algorithm updates the set of executed subtasks σ'_f and calculates the new progress value $P(\sigma_f)$

4.3 Analysis and Dynamic Tuning of FDM-TS Algorithm

The robot selects the task with maximal utility; if all utilities are zero the robot terminates its participation. The decision procedure has computational complexity linear in the number of tasks, $\mathcal{O}(m)$, since the number of fuzzy rules is constant.

A principal limitation of the presented FDM-TS is the *static* nature of its fuzzy membership functions and rule parameters. These parameters are defined offline and remain unchanged during the mission. While static parameters guarantee repeatability, they may degrade performance when task distributions, robot densities, or environmental conditions vary significantly over time. In particular, fixed membership shapes can cause suboptimal utility evaluations in phases where: (i) robot densities fluctuate rapidly, (ii) desired ratio of subtask $\tau(\sigma_i)$ change, or (iii) sensing noise patterns vary.

To address this limitation, FDM-TS is augmented with a dynamic tuning mechanism based on Particle Swarm Optimization (PSO). In this framework, PSO would iteratively adjust the parameters of the fuzzy membership functions during runtime, using the observed global allocation error as a fitness measure. Such an adaptive scheme would allow the decision system to continuously optimize its inference process, improving responsiveness and performance in highly dynamic and uncertain environments.

5. PARTICLE SWARM OPTIMIZATION FOR FDM-TS (PSO-FDM-TS)

This section outlines the optimization framework combining FDM-TS with Particle Swarm Optimization (PSO). Task utility in FDM-TS is evaluated through rule-based reasoning using predicted complex task error $PCE(T,k)$ and observation of active robot $D(r)$ as input variables. In order to improve its performance at the mission level, the parameters of the membership functions are optimized using PSO. The optimization process aims to minimize the global mission error noted by $\Delta_{\text{PSO}}(T,\theta)$, where θ is the vector of PSO parameters, thereby aligning local fuzzy decisions with global task-allocation objectives.

5.1 Particle Swarm Encoding for FDM-TS

PSO-FDM-TS is defined based on two input variables: $PCE(T,k) \in [-1,1]$ and $D(r) \in [0,1]$, along with one output variable $u(t_{fj}) \in [0,1]$. Each variable is associated with a set of linguistic terms represented by trapezoidal or triangular membership functions. Optimizing these membership functions constitutes a parameter-tuning problem, formulated as a vector manipulated by the Particle Swarm Optimization (PSO) algorithm.

In general, the trapezoidal membership function, parameterized by $(a<b<c<d)$, is defined as:

$$\mu_{\text{Trap}}(x;a,b,c,d)=\begin{cases}0, & x\le a,\\ \dfrac{x-a}{b-a}, & a<x\le b,\\ 1, & b<x\le c,\\ \dfrac{d-x}{d-c}, & c<x\le d,\\ 0, & x\ge d.\end{cases} \tag{23}$$

Similarly, the triangular function, defined by $(a<b<c)$, is expressed as:

$$\mu_{\text{Tri}}(x;a,b,c)=\begin{cases}0, & x\le a \text{ or } x\ge c,\\ \dfrac{x-a}{b-a}, & a<x\le b,\\ \dfrac{c-x}{c-b}, & b<x<c.\end{cases} \tag{24}$$

Fuzzy Variables and Linguistic Terms: The first input variable $x_1 = PCE(T,k)$ is described using five linguistic terms: BNE, ANE, SNE, ZE, and POS:

$$\mu_{\text{BNE}}(x_1)=\mu_{\text{Trap}}(x_1;-1,E_{b1},E_{c1},E_{d1}), \tag{25a}$$
$$\mu_{\text{ANE}}(x_1)=\mu_{\text{Trap}}(x_1;E_{a2},E_{b2},E_{c2},E_{d2},), \tag{25b}$$
$$\mu_{\text{SNE}}(x_1)=\mu_{\text{Tri}}(x_1;E_{a3},E_{b3},E_{c3}), \tag{25c}$$
$$\mu_{\text{ZE}}(x_1)=\mu_{\text{Tri}}(x_1;E_{a4},E_{b4},E_{c4}), \tag{25d}$$
$$\mu_{\text{POS}}(x_1)=\mu_{\text{Trap}}(x_1;0.0001,0.05,1,1). \tag{25e}$$

The second input variable $x_2 = D(r)$ contains three linguistic terms:

$$\mu_{\text{LD}}(x_2)=\mu_{\text{Trap}}(x_2;0,D_{b1},D_{c1},D_{d1}), \tag{26a}$$
$$\mu_{\text{MD}}(x_2)=\mu_{\text{Tri}}(x_2;D_{a2},D_{b2},D_{c2}), \tag{26b}$$
$$\mu_{\text{ID}}(x_2)=\mu_{\text{Trap}}(x_2;D_{a3},D_{b3},D_{c3},1). \tag{26c}$$

The output variable $y = u(t_{fj})$ is represented with four linguistic terms:

$$\mu_{\text{ZU}}(y)=\mu_{\text{Trap}}(y;0,U_{b1},U_{c1},U_{d1}), \tag{27a}$$
$$\mu_{\text{NU}}(y)=\mu_{\text{Trap}}(y;U_{a2},U_{b2},U_{c2},U_{d2}), \tag{27b}$$
$$\mu_{\text{IU}}(y)=\mu_{\text{Trap}}(y;U_{a3},U_{b3},U_{c3},U_{d3}), \tag{27c}$$
$$\mu_{\text{VIU}}(y)=\mu_{\text{Trap}}(y;U_{a4},U_{b4},U_{c4},1). \tag{27d}$$

Vector of Optimizable Parameters: All fuzzy parameters are collected into a global vector:

$$\theta=\big(\theta(G),\theta(D),\theta(U)\big)\in\mathbb{R}^{36}, \tag{28}$$

with:

$$\theta(G)=\big(E_{b1},E_{c1},E_{d1},E_{a2},E_{b2},E_{c2},E_{d2},E_{a3},E_{b3},E_{c3},E_{a4},E_{b4},E_{c4}\big) \tag{29a}$$
$$\theta(D)=\big(D_{b1},D_{d1},D_{c1},D_{a2},D_{b2},D_{c2},D_{a3},D_{b3},D_{c3}\big) \tag{29b}$$
$$\theta(U)=\big(U_{b1},U_{c1},U_{d1},U_{a2},U_{b2},U_{c2},U_{d2},U_{a3},U_{b3},U_{c3},U_{d3},U_{a4},U_{b4},U_{c4}\big) \tag{29c}$$

Thus, each PSO particle is represented by a particular instance of the vector θ.

5.2 PSO Dynamics

In PSO, each particle i is characterized by a position $x_i(t)$ that corresponds to a parameter vector θ. Its dynamics are governed by:

$$v_i(t+1)=\omega v_i(t)+c_1r_1\big(pbest_i-x_i(t)\big)+c_2r_2\big(gbest-x_i(t)\big), \tag{30}$$
$$x_i(t+1)=x_i(t)+v_i(t+1), \tag{31}$$

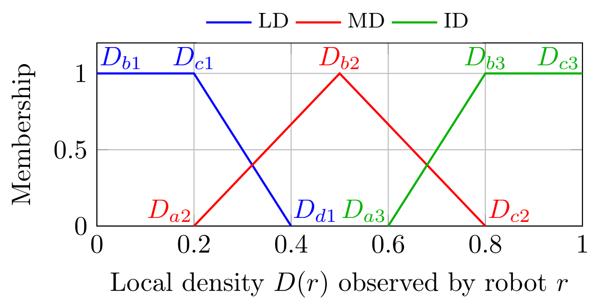


Fig. 4. Codification of the membership function parameters for the input variable $D(r)$

where ω denotes the inertia weight, c_1 and c_2 are the cognitive and social acceleration coefficients, and $r_1, r_2 \in [0.1]$ are uniformly distributed random variables. Here, $pbest_i$ represents the best local position reached by particle i, and $gbest$ is the best global position of the swarm.

5.3 Fitness Function

In the proposed framework, the fitness function returned by each particle does not measure instantaneous utility u_{t_k} for elementary tasks k but rather the *global error* $\Delta(T)$ accumulated over the entire mission, as expressed in eq. (8). Formally,

$$\Delta_{\text{PSO}}(T, \theta) = \Delta(T) \tag{32}$$

where $\Delta(T)$ is obtained by the FDM-TS after its membership functions have been parameterized by the vector θ.

The optimization problem is then formulated as

$$\theta^\star = \arg\min_{\theta\in\Omega}[\Delta_{\text{PSO}}(T, \theta)], \tag{33}$$

with: Ω denoting the set of feasible solutions.

In this formulation, the PSO algorithm directly searches for the fuzzy parameterization that minimizes mission-wide error, thereby aligning local task-utility decisions with global performance objectives.

5.4 Constraints on Fuzzy Parameters

The feasible set Ω is determined by structural and semantic constraints on the fuzzy membership functions. Each trapezoidal function defined by (a, b, c, d) must satisfy the ordering

$$a < b < c < d, \qquad (a, b, c, d) \in \theta, \tag{34}$$

while triangular functions defined by (a, b, c) must satisfy

$$a < b < c, \qquad (a, b, c) \in \theta. \tag{35}$$

All breakpoints must remain within the physical domain of the corresponding variables:

$$-1 \le Ea_i, Eb_i, Ec_i, Ed_i \le 1, \tag{36}$$

$$0 \le Da_i, Db_i, Dc_i \le 1, \tag{37}$$

$$0 \le Ua_i, Ub_i, Uc_i, Ud_i \le 1. \tag{38}$$

Furthermore, non-degeneracy constraints enforce a minimum separation $\varepsilon > 0$ between consecutive breakpoints:

$$(b - a) \ge \varepsilon, \quad (c - b) \ge \varepsilon, \quad (d - c) \ge \varepsilon. \tag{39}$$

To preserve the intended semantics of the fuzzy terms, the centroids of the sets are ordered consistently:

$$m_{\text{BNE}} < m_{\text{ANE}} < m_{\text{SNE}} < m_{\text{ZE}} < m_{\text{POS}}, \tag{40}$$

$$m_{\text{LD}} < m_{\text{MD}} < m_{\text{ID}}, \tag{41}$$

$$m_{\text{ZU}} < m_{\text{NU}} < m_{\text{IU}} < m_{\text{VIU}}. \tag{42}$$

Finally, coverage and overlap constraints ensure that the membership functions jointly span the entire domain without gaps, while maintaining smooth transitions between adjacent terms.

5.5 PSO-FDM-TS Algorithm

Figure 5 illustrates how the optimization process (PSO) interfaces with the fuzzy decision-making layer FDM-TS.

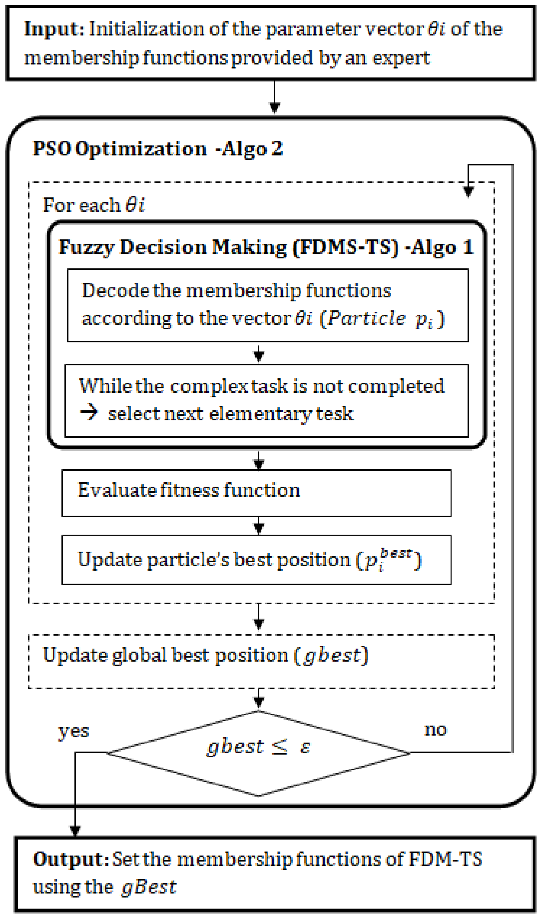


Fig. 5. Interactions between the optimization process (PSO) and FDM-TS

Algorithm 2 begins by initializing a swarm of particles, each representing a candidate set of fuzzy membership function parameters θ. The fitness of each particle is evaluated as the global mission error $\Delta_{\text{PSO}}(T, \theta)$ obtained after executing the complex task with the FDM-TS strategy. At each iteration, the particles update their velocities and positions based on their personal best and global best solutions. The process continues until the admissible error threshold ε is reached according to (19); or a maximum number of iterations are completed. Finally, the best parameter vector $\theta^\star$ is returned.

5.6 Computational Complexity Analysis of PSO-FDM-TS

A key advantage of the proposed PSO-FDM-TS scheme lies in its independence from the number of robots. Since the optimization is fully decentralized, each robot performs the PSO-based tuning locally. No centralized optimizer is required, which means that all robots execute the algorithm in parallel. As a result, the computational load does not scale with the size of the robotic team, and the time complexity depends only on the parameters of the optimization process.

The computational cost of PSO-FDM-TS depends on the number of particles N_p, the dimensionality d of the fuzzy parameter vector, the maximum number of iterations $I_{\max}$, and the cost C_f of evaluating the fitness function $\Delta_{\text{PSO}}(T, \theta)$ for one particle given by FDM-TS. Updating

Algorithm 2 PSO-FDM-TS Optimization

Input: Expert parameters θ^{exp}, admissible error ε, swarm size N_p, max iterations I_{max}
Output: Optimized fuzzy parameters $\theta^{\star}$
Initialize swarm of N_p particles with random positions $\theta_i(0)$ around θ^{exp}
Initialize velocities $v_i(0)$ and set personal bests $pbest_i \leftarrow \theta_i(0)$
Evaluate fitness of each particle: $\Delta_{\text{PSO}}(T, \theta_i)$
Set global best $gbest \leftarrow \arg\min_i \Delta_{\text{PSO}}(T, \theta_i)$
for $t = 1$ to I_{max} **do**
  for each particle i **do**
    Update velocity $v_i(t+1)$, (Eq.30)
    Update position $\theta_i(t+1)$, (Eq.31)
    Evaluate fitness $\Delta_{\text{PSO}}(T, \theta_i(t+1))$, (Eq.32)
    if $\Delta_{PSO}(T, \theta_i(t+1)) < \Delta_{PSO}(T, pbest_i)$ **then**
      $pbest_i \leftarrow \theta_i(t+1)$
    end if
    if $\Delta_{PSO}(T, \theta_i(t+1)) < \Delta_{PSO}(T, gbest)$ **then**
      $gbest \leftarrow \theta_i(t+1)$
    end if
  end for
  if $\Delta_{PSO}(T, gbest) \leq \varepsilon$ **then**
    break
  end if
end for
Return: $\theta^{\star} \leftarrow gbest$

the velocity and position of a particle requires vector operations of size d, resulting in a per-particle cost of:

$$O(d + C_f). \tag{43}$$

Since all N_p particles perform these operations at each iteration, the total complexity per iteration is:

$$O(N_p.(d + C_f)). \tag{44}$$

Considering the complete optimization process during $I_{\max}$ iterations, the overall time complexity of the algorithm becomes:

$$O(N_p \cdot I_{\max} \cdot (d + C_f)). \tag{45}$$

In many practical cases, the fitness evaluation dominates the computation time ($C_f \gg d$). Under such conditions, the complexity can be simplified as:

$$O(N_p \cdot I_{\max} \cdot C_f). \tag{46}$$

6. EXPERIMENTS AND RESULTS

6.1 Simulation environment

To experimentally demonstrate the role of PSO-FDM-TS, the FDM-TS algorithm was re-implemented, and the second algorithm was developed in Java using the Eclipse IDE (version 4.35.0). To simulate the Heterogeneous Multi-Robot System (HMRS), a realistic simulation environment was employed. The experiments were conducted using the MobileSim Simulation Environment (Omron, 2018a), which provides physically realistic models of mobile robots, such as the Pioneer P3-DX with two-wheel drive and the P3-AT with four-wheel drive, along with their operating environments. The Java ARIA packages (Omron, 2018b) were employed to supply various tools for programming high-level control actions governing Pioneer P3-DX movements, as well as for developing a Java-based client application for complex task descriptions. Moreover, the apper3Basic environment (Omron, 2018c) was utilized to design a realistic environment map with fixed obstacles in a 40 × 25 m enclosed space as shown in Fig. 6. Although the robots appear visually identical in the MobilSim environment, their underlying capabilities differ, reflecting distinct functional roles and performance characteristics within the simulated system.

6.2 Warehouse Transportation Scenario

The proposed approaches were evaluated in the context of goods transportation within a supply warehouse, where autonomous robots are tasked with delivering heterogeneous items from multiple fixed supply stations to a single assembly station, following predefined demand ratios. Robots can access information about resource requirements and the progress of the assembly task only when located in the assembly area, distant from the supply stations. The selection of elementary tasks, defined as the transport of a single resource from a supply station to an assembly station, is restricted to robots physically present at a supply station. The decision regarding which type of resource to transport is based on previously stored observations from the assembly area combined with real-time data from supply stations. Each elementary task is non-preemptive, and the mission is considered complete once the assembly station's requirements are satisfied.

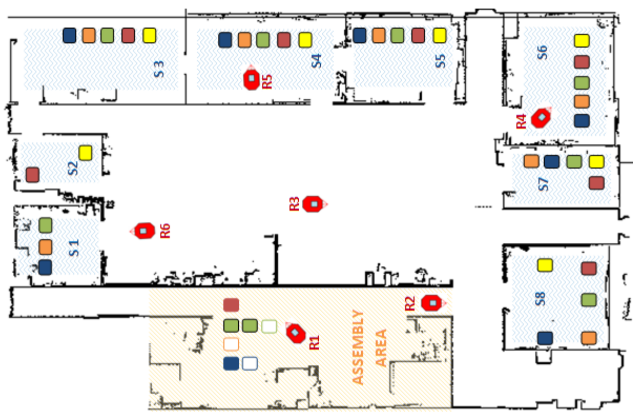


Fig. 6. Simulation snapshot of the transportation scenario with six Pioneer P3DX robots at time $t = t'$.

The experimental setup consists of six heterogeneous mobile robots. The complex assembly task is decomposed into four subtasks ($\sigma_1 \ldots \sigma_4$) requiring a total of N = 7 elementary tasks as shown in Fig. 6 at the assembly area. Subtasks contain respectively $|\sigma_1| = 2$, $|\sigma_2| = 1$, $|\sigma_3| = 3$, $|\sigma_4| = 1$ elementary tasks. Robots select tasks only when located at the supply stations, whereas global progress can be observed exclusively when entering the assembly zone. This ensures a realistic separation between local perception and global information. All these parameters are représented in table 2.

At time t', all six robots remain active, but only three subtasks are still pending, as summarized in Table 3.

The observations of the complex task differ across robots. For example, at time t', robots $R1$ and $R2$ update their

Table 2. Configuration of the complex task

nb robot	$\lvert\sigma_1\rvert$	$\lvert\sigma_2\rvert$	$\lvert\sigma_3\rvert$	$\lvert\sigma_4\rvert$
6	2	1	3	1

Table 3. Desired ratios, achieved progress, and relative errors at $t = 0$ and $t = t'$.

Subtask	$t = 0$			$t = t'$		
	$\tau(\sigma_i)$	$P(\sigma_i)$	$\delta(\sigma_i)$	$\tau(\sigma_i)$	$P(\sigma_i)$	$\delta(\sigma_i)$
σ_1	0.29	0.00	-0.29	0.29	0.14	-0.14
σ_2	0.14	0.00	-0.14	0.14	0.00	-0.14
σ_3	0.43	0.00	-0.43	0.43	0.29	-0.14
σ_4	0.14	0.00	-0.14	0.14	0.14	0.00
Global Error $\Delta(T)$	**-1.00**			**-0.43**		

view of the global progression task state after completing their respective actions in the assembly area, while robots $R3..R6$ rely on delayed information from $t = t' - 1$. Moreover, each robot R_i possesses distinct local observations of active robots at time t' (Table 4). This emphasizes the role of the fuzzy predictor, which determines the optimal allocation strategy π_i when robots arrive at the supply station.

Table 4. Local observations of active robots at $t = t' - 1$ and $t = t'$.

Robot	$t = t' - 1$	$t = t'$
R_1	0	1
R_2	1	0
R_3	2	2
R_4	5	5
R_5	3	3
R_6	4	4

According to the FDM-TS rules, the local observation of robot R_i is used to reduce the importance of task t_k when the density of active robots is high and the value of $\Delta'(T, k)$ is very low, since there is a strong likelihood that another robot will select this task t_f. Conversely, the predictor assigns greater importance to another task t_i, for which $\Delta'(T, i) > \Delta'(T, k)$, but which has a lower probability of being selected. Thus, in the scenario illustrated in Fig. 6 at time t', robot R_5, with a local observation of three active robots, selects an elementary task from subtask σ_1, whereas robot R_4, with a local observation of five active robots, selects an elementary task belonging to subtask σ_3.

6.3 Comparison of OCTD, FDM-TS, and PSO-FDM-TS

To further evaluate the effectiveness of the proposed approach, a comparative study is conducted involving three different models for generating the allocation strategy π_i. The first model is based exclusively on the Observation of the Complex Task Decomposition (OCTD), without the integration of the fuzzy predictor. The second model is based on the FDM-TS mechanism, while the third integrates parameter optimization through PSO-FDM-TS.

These models are systematically assessed through three series of experiments:

- **Case 1 – Task heterogeneity:** The number of robots is fixed to 6, while the number of subtasks σ_i ranges from 1 to 12 with 1 to 80 elementary tasks. The distribution of desired ratios $\tau(\sigma_i)$ is determined according to Fig. 7.
- **Case 2 – Heterogeneity of the MRS:** The complex task is fixed to six subtasks with desired ratios identical to Case 1, while the number of robots is varied from 1 to 12.
- **Case 3 – Scalability:** A large-scale scenario is considered with 50 subtasks, representing a total of 1200 elementary tasks. The number of robots is varied from 1 to 100. In this series, the FDM-TS is parameterized with optimized membership functions obtained from the PSO-FDM-TS method.

6.4 Results and Discussion

Task Heterogeneity. As illustrated in Fig. 8, the global error increases sharply with the number of subtasks when using the OCTD baseline, highlighting its poor scalability in heterogeneous task distributions. The FDM-TS approach mitigates this trend by leveraging fuzzy reasoning to balance task selection, resulting in substantially lower errors. However, residual fluctuations remain due to the static nature of its membership functions. In contrast, PSO-FDM-TS maintains consistently low error levels, even when the number of subtasks reaches twelve. This confirms the superior robustness of the optimized model in handling task diversity.

Heterogeneity of the MRS. The impact of increasing the number of robots is shown in Fig. 9. The OCTD baseline suffers from redundancy, as multiple robots frequently select the same task, leading to large allocation errors. The FDM-TS model alleviates this issue by reducing conflicts through fuzzy-based task utilities, but error accumulation remains non-negligible at higher robot counts. By con-

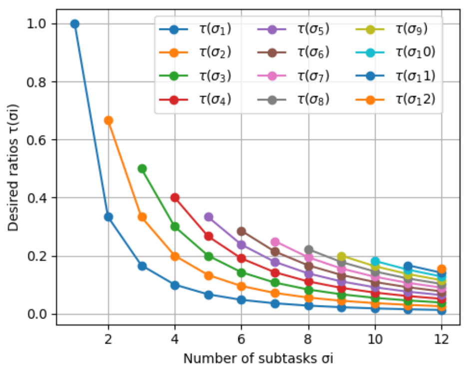


Fig. 7. Desired distribution ratios $\tau(\sigma_i)$.

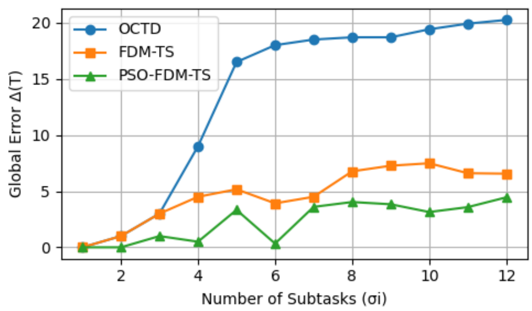


Fig. 8. Comparison of global error with increasing number of subtasks.

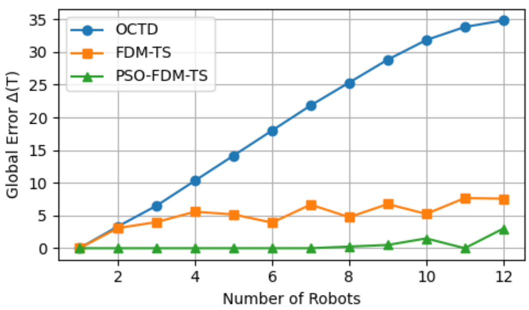


Fig. 9. Comparison of global error with increasing number of robots.

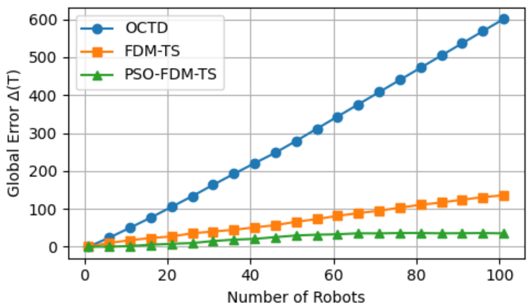


Fig. 10. Comparison of global error in large-scale scenarios.

trast, PSO-FDM-TS achieves near-zero global error for most scenarios, demonstrating its ability to adaptively tune fuzzy parameters and prevent redundancy in a fully decentralized setting.

Scalability. The large-scale scenario with 50 subtasks and up to 100 robots (Fig. 10) emphasizes the scalability limits of the three models. OCTD rapidly becomes ineffective, with error values escalating uncontrollably as the system grows. The FDM-TS approach maintains a more acceptable error profile, but still experiences degradation as both the number of subtasks and robots increase. PSO-FDM-TS significantly reduces the global error compared to both baselines, showing its effectiveness at large scales. Nevertheless, some error growth remains unavoidable due to the inherent combinatorial complexity of the problem.

Analysis of Relative Errors. In addition to global error evaluation, a detailed analysis of relative errors across subtasks provides deeper insights into the allocation accuracy as shown in Fig.11. The OCTD baseline exhibits irregular and high relative errors, indicating poor proportionality between the achieved and desired task ratios. The FDM-TS model significantly improves proportional allocation, yet relative errors remain unevenly distributed, especially in scenarios with fluctuating robot density. By contrast, PSO-FDM-TS achieves the lowest and most balanced relative errors, maintaining a closer match to the predefined task ratios across all subtasks. This is further illustrated by the average relative error in Fig.12, which demonstrates that PSO-based optimization not only reduces global mission error but also improves fairness in task distribution.

Discussion. The comparative evaluation highlights the key strengths of the proposed approaches. The integration of fuzzy reasoning in FDM-TS already improves allocation robustness compared to the OCTD baseline, while the PSO-FDM-TS model further enhances performance by adaptively tuning membership functions, yielding consistently lower global and relative errors across diverse scenarios. This confirms the effectiveness of combining fuzzy logic with evolutionary optimization in achieving fair and scalable task distribution in heterogeneous multi-robot systems. Nevertheless, certain limitations remain, particularly in large-scale deployments where residual error growth persists due to the combinatorial nature of the problem, and where the absence of inter-robot communication may restrict coordination efficiency.

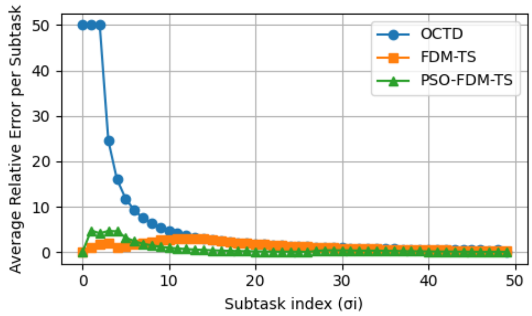


Fig. 11. Evolution of average relative error per subtask σ_i.

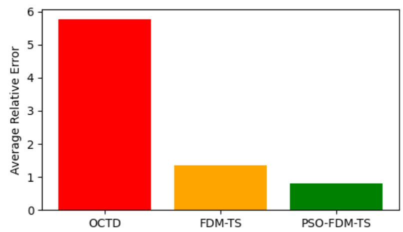


Fig. 12. Average relative error across models.

These findings pave the way for the concluding section, where perspectives for further research are outlined.

7. CONCLUSION

This work addressed the challenge of dynamic task allocation in heterogeneous multi-robot systems by introducing and comparing three models: OCTD, FDM-TS, and PSO-FDM-TS. The results demonstrated that the OCTD baseline suffers from scalability issues and high allocation errors, particularly in heterogeneous with large-scale scenarios. By incorporating fuzzy reasoning, FDM-TS improved robustness and reduced redundancy in execution of elementary tasks, although its static membership functions limited adaptability. The proposed PSO-FDM-TS model overcame these shortcomings by optimizing the fuzzy parameters through evolutionary learning, thereby achieving consistently lower global and relative errors across all test cases.

Despite these improvements, certain limitations remain. In large-scale scenarios, residual error growth could not be fully eliminated, reflecting the inherent combinatorial complexity of the problem. Moreover, the models were developed under the assumption of limited or no inter-robot communication, which may restrict coordination in highly dynamic environments.

Future work will explore hybrid optimization strategies that combine swarm intelligence with learning-based methods, as well as the integration of lightweight communication protocols to further enhance coordination efficiency. Additionally, applying the proposed approaches to real-world robotic platforms will provide further insights into their practical applicability and robustness under real-time operational constraints.

REFERENCES

Aliskan, I. (2025). The optimization-based fuzzy logic controllers for autonomous ground vehicle path tracking. *Engineering Applications of Artificial Intelligence*, 151(2), 110642. doi:10.1016/j.engappai.2025.110642.

Ching-Wei, C. and Wei-Yu, L. (2024). Task decomposition and multi-robot task allocation in exploration with bayesian networks. In *2024 Eighth IEEE International Conference on Robotic Computing (IRC)*, 80–83. IEEE.

Dahl, T.S., Matarić, M.J., and Sukhatme, G.S. (2009). Multi-robot task allocation through vacancy chain scheduling. *Robotics and Autonomous Systems*, 57(6), 674–687. doi:10.1016/j.robot.2008.12.001.

Do, H., Jang, J., and Kim, J. (2025). Heterogeneous multi-robot system mission planning with cooperative replenishment through data-driven rendezvous point selection. *Intelligent Service Robotics*, 18, 61–73. doi:0.1007/s11370-024-00568-9.

Gerkey, B.P. and Mataric, M.J. (2004). A formal analysis and taxonomy of task allocation in multi-robot systems. *The International Journal of Robotics Research*, 23(9). doi:10.1177/0278364904045564.

Gil, E.B., Rodrigues, G.N., Pelliccione, P., and Calinescu, R. (2023). Mission specification and decomposition for multi-robot systems. *Robotics and Autonomous Systems*, 163, 104386. doi:10.1016/j.robot.2023.104386.

Islam, M.R. and Akhtar, M.N. (2017). Fuzzy logic based task allocation in ant colonies under grid computing. In *2017 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, 22–27.

Jaume-Martin, G., Antich, J., Guerrero, J., and Valero, O. (2024). An application of fuzzy sets to optimal task-allocation problem. In *Intelligent and Fuzzy Systems*, 385–393. Springer Nature Switzerland.

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, 1942–1948. doi:10.1109/ICNN.1995.488968.

Mayya, S., D'antonio, D.S., Saldaña, D., and Kumar, V. (2021). Resilient task allocation in heterogeneous multi-robot systems. *IEEE Transactions on Robotics*, 6(2), 1327–1334. doi:10.48550/arXiv.2009.04593.

Mulani, A. and Ramachandran, S. (2023). Multi-robot task allocation methods: A fuzzy optimization approach. *Information Sciences*, 648, 119508. doi:10.1016/j.ins.2023.119508.

Oliveira, G.S., Röning, J., Plentz, P.D.M., and Carvalho, J.T. (2022). Efficient task allocation in smart warehouses with multi-delivery stations and heterogeneous robots. In *Proceedings of the 2022 25th International Conference on Information Fusion (FUSION)*, 1–8. IEEE.

Prauzek, M., Gaiova, K., Kucova, T., and Konecny, J. (2025). Fuzzy energy management strategies for energy harvesting iot nodes based on a digital twin concept. *Future Generation Computer Systems*, 166, 107717. doi:10.1016/j.future.2025.107717.

Rafik, E., Noureddine, B., and Redha, Rebhiand Mohamed, R. (2022). Optimal anfis-fopid with backstepping controller design for wheeled mobile robot control. *Journal of Control Engineering and Applied Informatics*, 24(2).

Rechache, K., Teggar, H., and Boufera, F. (2024). Efficiency of fuzzy task assignment in heterogeneous multi-robot system. *Journal of Control Engineering and Applied Informatics*, 26(3).

Rizk, Y., Awad, M., and Tunstel, E. (2019). Cooperative heterogeneous multi-robot systems: A survey. *ACM Computing Surveys*, 52(2), 1–31. doi:10.1145/3303848.

Roy, N., Beauthier, C., and Mayer, A. (2022). Setup of a new adaptive fuzzy particle swarm optimization algorithm. In *2022 IEEE Congress on Evolutionary Computation (CEC)*, 1–8.

Teggar, H., Mohamed, S., and Fatima, D. (2021a). Accuracy in parallel dynamic task allocation for multi-robot systems under fuzzy environment. *International Journal of Fuzzy System Applications*, 10(2). doi:10.4018/IJFSA.2021040101.

Teggar, H., Mohamed, S., and Fatima, D. (2021b). Fuzzy predictor for parallel dynamic task allocation in multi-robot systems. *International Journal of Computer Aided Engineering and Technology*, 15(1), 12–31. doi:10.1504/IJCAET.2021.115944.

Thi, T., Van, B., and Hung, L. (2022). Stabilizing the flexible convergence of swarm robot by fuzzy attraction and repulsion. *Journal of Control Engineering and Applied Informatics*, 24(4).

Valero, O., Antich, J., Tauler-Rosselló, A., Guerrero, J., Miñana, J.J., and Ortiz, A. (2023). Multi-robot task allocation methods: A fuzzy optimization approach. *Information Sciences*, 648, 119508. doi:10.1016/j.ins.2023.119508.

Verma, A., Gautam, A., Dutta, A., Shekhawat, V.S., and Mohan, S. (2025). Cf-hmrta: Coalition formation for heterogeneous multi-robot task allocation. *Journal of Intelligent & Robotic Systems*, 111(18). doi:10.1007/s10846-025-02287-4.

Xidias, E. and Zacharia, P. (2024). Balanced task allocation and motion planning of a multi-robot system under fuzzy time windows. *Engineering Computations*, 41(5), 1301–1326. doi:10.1108/EC-09-2023-0612.

Zadeh, L.A. (1965). Fuzzy sets. *Information and Control*, 8(3), 338–353. doi:S0019-9958(65)90241-X.

Zhao, T. and Wang, Y. (2013). Market based multi-robot coordination for a cooperative collecting and transportation problem. In *2013 Proceedings of IEEE Southeastcon*, 1–6. IEEE. doi:10.1109/SECON.2013.6567397.