Towards Learning-based Distributed Task Allocation Approach for Multi-Robot System

Zakaria Chekakta*, Nabil Aouf*, Shashank Govindaraj†, Fabio Polisano† and Geert De Cubber‡
* City, University of London, United Kingdom
†Space Applications Services NV/SA, , Brussels, Belgium
‡ Royal Military Academy, Brussels, Belgium
Emails: zakaria.chekakta@city.ac.uk, nabil.aouf@city.ac.uk, shashank.govindaraj@spaceapplications.com, fabio.polisano@spaceapplications.com, geert.de.cubber@mil.be

***Abstract*—This paper introduces a novel application of Graph Convolutional Networks (GCNs) for enhancing the efficiency of the Consensus-Based Bundle Algorithm (CBBA) in multi-robot task allocation scenarios. The proposed approach in this research lies in the integration of a learning-based strategy to approximate the heuristic methods traditionally used for scoring in the CBBA framework. By employing GCNs, the proposed methodology aims to learn and predict the score function, which is crucial for task allocation decisions in multi-robot systems. This approach not only streamlines the allocation process but also potentially improves the accuracy and efficiency of task distribution among robots. The paper presents a detailed exploration of how GCNs can be effectively tailored for this specific application, along with results demonstrating the advantages of this learning-based approach over conventional heuristic methods in various simulated multi-robot task allocation scenarios.**

***Keywords*—Task Allocation, Multirobot System, Distributed Algorithms, Graph Convolutional Neural Networks**

I. Introduction

The task allocation problem aims to find a globally feasible allocation of tasks to agents while optimizing one or more objectives. For Multi-Robot Systems (MRS) with varied capabilities, two main challenges arise: the high computational complexity of traversal algorithms and the limitations of centralized algorithms, including reduced task range and single point of failure risk [1], [2]. To address these, heuristic algorithms are used as a more efficient, though not always optimal, alternative to traversal algorithms. The effectiveness of a given heuristic is dependent on various factors including the constraints and parameters of the problem being solved and the objective being optimized [3]. Additionally, distributed algorithms replace centralized ones, enhancing task range and system robustness by distributing decision-making across agents. Distributed consensus-based algorithms can solve task allocation problems in a cooperative planning process consisting of two phases [4]. In the first phase, an agent constructs a schedule of selected tasks through an internal decision-making process. This process has previously been referred to as a utility function [5], a score function [6], or an objective function [7]. In the second phase, agents communicate bids on their selected task allocation and resolve conflicts by assigning tasks to the agents with the highest bids. Agents perform one task at a time, and each agent can be assigned multiple tasks that they execute based on a schedule. Travel times, task durations, task deadlines, and fuel constraints are factors. Finding the optimal solution to this task allocation problem in real-time environments becomes computationally unfeasible as the number of tasks/agents grows. However, distributed algorithms assume ideal communication conditions and rely on consensus for consistent situational awareness (SA). Current state-of-the-art consensus-based task allocation algorithms incorporate heuristics into agent score functions in order to optimize a given objective. While extensive research has been done in the area of multi-agent learning of optimal policies [8], [9]. Each solution involves trade-offs between efficiency, optimality, and robustness [10], [11], [12], [13].

An alternative approach to resolving this dilemma is the implementation of the auction algorithm. In the auction algorithm, agents bid on individual tasks, and a central system or designated agent acts as an auctioneer to select the winning bids. The bundle algorithm simplifies this by having agents bid on groups of tasks, or bundles, rather than single tasks. While both methods offer dynamic and potentially efficient task allocation, they may not be as robust as consensus algorithms in adapting to changes in communication networks. However, traditional auction algorithms are generally more computationally efficient compared to consensus algorithms, which excel in robustness but may lack in speed, especially in large-scale systems. The choice between these methods depends on the MRS's specific needs for adaptability, efficiency, and communication robustness.

The Consensus-Based Bundle Algorithm (CBBA) [4] discussed in this paper is a hybrid approach for task allocation in MRS, combining auction-based methods and consensus algorithms. It uses auctions to distribute tasks among robots and employs a consensus mechanism to resolve any conflicts arising from overlapping bids or dependencies. CBBA stands out for its efficient convergence, quickly reaching a stable state of task allocation. Additionally, it guarantees at least 50% optimality in its solutions, when the bidding price has the diminishing marginal gain (DMG) property [14]. Balancing speed and efficiency with a reasonable level of accuracy. The CBBA effectively addresses the challenges of distributed task allocation by combining the dynamic nature of auctions with the conflict resolution capabilities of consensus algorithms.

On the other hand, Graph Convolutional Networks (GCNs) exhibit exceptional capabilities in their application to large-scale robotic teams. These networks showcase exceptional performance and demonstrate an impressive ability to generalize across a wide array of complex tasks [15]. This includes sophisticated applications like coordinated flocking, advanced navigation strategies, and precise control mechanisms [16], [17], [18], [19]. The proficiency of GCNs in seamlessly adapting to these diverse and challenging tasks underscores their pivotal role in revolutionizing the capabilities of multi-robot systems.

In this study, we introduce an advanced AI-enhanced Consensus-Based Bundle Algorithm (AI-CBBA), tailored specifically for optimizing task allocation for multi-robot systems. The paper begins by constructing a formal model for the task allocation problem. This is followed by an exploration of the consensus-based bundle algorithm, providing a brief understanding of its mechanisms. Building upon this, the study delves into a set of heuristic methods designed for the original CBBA, enhancing its efficiency and effectiveness. A pivotal component of our approach is the integration of a GCN-based architecture to predict the score function. The paper finalizes with a presentation of results and a detailed discussion, where the performance of our proposed algorithm is compared against the existing state-of-the-art solutions, demonstrating its potential in managing complex task allocations. The proposed algorithm has been used for Explosive Ordnance Disposal mission [20].

II. Learning-based Distributed Task Allocation

A. Problem Formulation

In this section, we introduce a mathematical formulation for the task allocation problem aimed at maximizing the total score. We propose a binary integer programming model where the decision variable x_{ij} signifies whether task j is assigned to agent i, and the score function $S_{ij}(x_i, \eta_i)$ measures the utility of such an assignment.

The formulation is bound by constraints ensuring each agent can undertake no more than a set number of tasks K_i, each task may only be allocated to one agent, and the total number of tasks assigned does not exceed the number of available tasks or the cumulative maximum capacities of the agents. Table I outlines the parameters used in our model.

$$
\begin{aligned}
\underset{x}{\text{maximize}} \quad & \sum_{i=1}^{N}\sum_{j=1}^{M} S_{ij}(x_i,\eta_i)x_{ij} \\
\text{subject to} \quad & \sum_{j=1}^{M} x_{ij} \le K_i, \quad \forall i \in I_a, \\
& \sum_{i=1}^{N} x_{ij} \le 1, \quad \forall j \in I_t, \\
& \sum_{i=1}^{N}\sum_{j=1}^{M} x_{ij} = \min\{M, \sum_{i=1}^{N} K_i\}, \\
& x_{ij} \in \{0,1\}, \quad \forall (i,j) \in I_a \times I_t.
\end{aligned} \tag{1}
$$

TABLE I

Parameters Description

Parameter	Description
N	Number of agents
M	Number of tasks
$x_{ij} = 1$	If agent i is assigned to task j and 0 otherwise
$x_i \in \{0,1\}^M$	Vector with the jth element as x_{ij}
$I_a = \{1, \cdots, N\}$	Set of agents
$I_t = \{1, \cdots, M\}$	Set of tasks
$\eta_i \in (I_t \cup \emptyset)^{K_i}$	Ordered set of tasks assigned to agent i
$\emptyset$	Empty set symbolizing no task
$S_{ij}(x_i, \eta_i)$	Score function of assigning task j to agent i
K_i	Maximum number of tasks to agent i

In the CBBA framework, the task allocation process comprises two distinct phases. The initial phase is dedicated to the generation of bids for tasks (Path Planning) by individual agents, while the second phase, known as conflict resolution, centers on the exchange of information among agents regarding their bids and the provisional allocation of tasks.

a) Phase I: Task Planning (Bundle Building): The path planning algorithm presented in Algorithm 1 outlines the procedure for constructing an optimized task bundle for an individual agent within a multi-agent system. At the beginning of each iteration, the agent's task state is initialized with the current bundle, owned tasks, potential task set, and task sequence. The core of the algorithm lies in the while-loop, which ensures that only the maximum permissible number of tasks K_i are considered for bundle construction. Within the loop, the algorithm performs a sequence of steps to evaluate which tasks from the set of unallocated tasks $I_t - \zeta_i(t)$ should be included in the agent's bundle.

For each of these tasks, a score $\tilde{S}_{ij}$ is computed,

$$
\tilde{S}_{ij}[\zeta_i] = \begin{cases} 0, & \text{if } j \in \zeta_i, \\ \max_{n \le |\eta_i|}\{\Theta_i^{\eta_i \oplus_n \{j\}} - \Theta_i^{\eta_i}\}, & \text{otherwise.} \end{cases} \tag{2}
$$

where $\Theta_i^{\eta_i}$ is the total reward minus the cost for the sequence η_i, and "$\oplus_n$", denotes the operation that inserts task j at the n-th position in the sequence η_i.

The decision to add a task to the bundle is contingent upon the score $\tilde{S}_{ij}$ exceeding a dynamic threshold ω_{ij}, which reflects the task's relative value and competitiveness against bids from other agents.

Following the scoring process, the algorithm identifies the task with the highest utility to be included in the agent's path and updates the task sequence and the set of potential tasks accordingly. Task ownership is reassigned to reflect this inclusion, and the associated thresholds are updated, which will influence subsequent iterations and allocations. The algorithm concludes its current iteration when the task bundle is solidified, meeting the capacity constraints of the agent. This iterative process is executed by each agent in a distributed manner, ensuring the system converges to a consistent global assignment that maximizes the overall score functions of all participating agents, thereby optimizing the allocation of tasks across the agent network.

Algorithm 1 Path Planning Algorithm for agent-i/iteration $(t+1)$

Process: Construct Bundle$(\nu_i(t),\ \sigma_i(t),\ \zeta_i(t),\ \eta_i(t))$
$\nu_i(t+1) = \nu_i(t)$
$\sigma_i(t+1) = \sigma_i(t)$
$\zeta_i(t+1) = \zeta_i(t)$
$\eta_i(t+1) = \eta_i(t)$
while $|\zeta_i| \geq K_i$ **do**
 $\tilde{S}_{ij}[\zeta_i] = \max_{n \leq |\eta_i|} \Theta_i^{\eta_i \oplus_n \{j\}} - \Theta_i^{\eta_i},\ j \in I_t - \zeta_i(t)$
 $\nu_{ij} = \mathbb{I}(\tilde{S}_{ij} > \omega_{ij}) \quad \forall j \in I_t$
 $J_i = \underset{j}{\operatorname{argmax}} \left(\tilde{S}_{ij}[\zeta_i] \times \nu_{ij} \right)$
 $n_{i,J_i} = \underset{n}{\operatorname{argmax}} \Theta_i^{\eta_i \oplus_n \{J_i\}}$
 $\eta_i = \eta_i \oplus_{n_{i,J_i}} \{J_i\}$
 $\zeta_i = \zeta_i \oplus_{end} \{J_i\}$
 $\omega_{iJ_i}(t+1) = \tilde{S}_{iJ_i}$
 $\sigma_{iJ_i}(t+1) = i$
End Process

b) Phase II: Conflict Resolution Procedure: During conflict resolution, agents communicate their bid values and the provisional winners for each task. The task is provisionally awarded to the agent with the highest marginal score for that task. An agent that has been outbid for a task must relinquish the task and any subsequent tasks in its bundle that were dependent on it.

This phase operates under the principle of Diminishing Marginal Gain (DMG), which posits that the marginal score for a task, denoted by $\tilde{S}_{ij}[\eta_i]$, should not increase with the addition of tasks to the agent's bundle. Formally, this is expressed as:

$$\tilde{S}_{ij}[\eta_i] \geq \tilde{S}_{ij}[\eta_i \oplus_{\text{end}} \{j\}], \tag{3}$$

where η_i is the current task bundle for agent i, and j is a new task being considered.

Convergence to a stable task allocation and a guarantee of at least 50% optimality are ensured by the CBBA under the DMG condition for the scoring function. Should the scoring function not naturally fulfill the DMG criterion, a warping mechanism is applied. The warping adjusts the score $\tilde{S}_{ij}[\eta_i]$ to:

$$\tilde{S}_{ij}[\eta_i] = \min\{\tilde{S}_{ij}[\eta]\}, \quad \forall \eta \subseteq \eta_i, \tag{4}$$

which assists in algorithm convergence when the natural scoring function lacks diminishing returns.

B. Learning-based Optimization

In recent years, learning-based optimization has emerged as a frontier in advanced computational methodologies, offering profound insights into complex problem-solving. This paradigm shift has been largely propelled by the advent and subsequent dominance of deep learning techniques, known for their unparalleled prowess in feature extraction and representation learning. Among the array of deep learning architectures, Convolutional Neural Networks (CNNs) have garnered widespread acclaim, especially for their performance in processing data characterized by a Euclidean or grid-like topology.

Despite their success, traditional CNNs encounter significant challenges when confronted with data embedded in non-Euclidean spaces, such as the intricate webs of social or information networks, where translation invariance is no longer a given. To bridge this gap, Graph Convolutional Networks (GCNs) have been introduced as a robust method for navigating the complex terrain of graph-structured data. GCNs have revolutionized our ability to tap into the rich vein of information contained within non-Euclidean domains, enabling the extraction of salient features that conventional methods would struggle to discern.

Considering a graph $G = (V, E)$ with V as the set of vertices and E as the edges denoting relationships, graph convolutions can be processed in either the spatial or spectral domains. Spatially, convolutions aggregate feature information from a node's local neighborhood directly, leveraging residual connections for deep memory across layers. Each vertex is equipped with its own neural network, and its activation in the k^{th} layer, denoted by $h_v^{(k)}$, is given by the equation:

$$h_v^{(k)} = \sigma \left(W^{(k)} x_v + \sum_{u \in \mathcal{N}(v)} \theta^{(k)} h_u^{(k-1)} \right)$$

where $W^{(k)}$ and $\theta^{(k)}$ are learned parameters for intra- and inter-nodal connections, respectively, and $\sigma(\cdot)$ represents a nonlinear activation function.

For the spectral domain, graph convolutions apply through the transformation of features into the Fourier space using eigendecomposition of the normalized graph Laplacian $L = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = U \Lambda U^T$. Here, U contains the eigenvectors, Λ is a diagonal matrix of eigenvalues, and the Fourier transformed features are $U^T x$. A filter parameterized by Θ operates on these transformed features, which is expressed as:

$$g'_\theta \star x = U g_\theta \Lambda U^T x$$

where g'_θ denotes the filtered signal. The adjacency matrix, with self-loops, is denoted by $\tilde{A} = A + I_N$, and the layer-wise propagation in the spectral GCN, which is utilized in this research, follows:

$$H^{(l+1)} = \sigma \left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)} \right)$$

In practical applications, such as multi-robot systems, graph structures capture the complexity of interactions within the system and between agents and environments. The agent-entity graph and task-entity graph encode these interactions. Through machine learning methodologies, specifically graph convolutional networks, we analyze these complex structures. For instance, we encode the position and attributes of tasks in a vector, apply a GCN to learn meaningful features from these relationships, and use the extracted features to understand the underlying data structure. The proposed distributed task allocation algorithm depicted in Figure 1 demonstrates the application of spectral GCNs for such feature extraction.

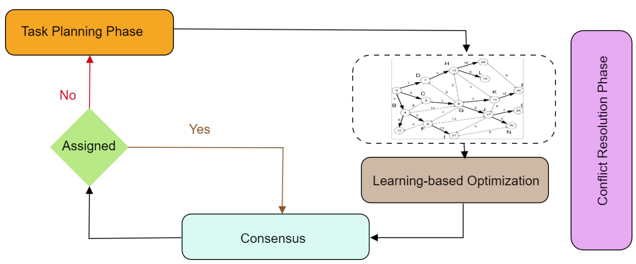

Fig. 1. Distributed Task Allocation

In alignment with the CBBA's foundational principles, we utilize $\xi_i(t)$ to signify the task sequence within agent i's bundle at time t, where ξ_i represents the ordered set of tasks. Notably, the sequence $\xi_i(t)$ does not necessarily correspond with the order of tasks within the bundle $\zeta_i(t)$. The path length, now interpreted as energy consumption, is represented by $D[\xi_i(t)]$. To adapt CBBA for scenarios with energy constraints, each agent commences by initializing their bundle $\zeta_i(t)$ to include starting point μ_i and a terminating point ν_i. For an agent i's current bundle $\zeta_i(t)$, the marginal utility $S_{ij}[\zeta_i(t)]$ of appending task j is conceptualized as the task's reward less the incremental energy cost, now expressed as:

$$S_{ij}[\zeta_i(t)] = \begin{cases} \gamma_{ij} - \Delta E_{ij}[\zeta_i(t)], & \text{if } \Delta E_{ij}[\zeta_i(t)] \leq E_i[\zeta_i(t)], \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

Here, $E_i[\zeta_i(t)]$ is the residual energy for agent i post-traversal of $\xi_i(t)$, while $\Delta E_{ij}[\zeta_i(t)]$ denotes the additional energy required if task j is to be included in $\xi_i(t)$.

In the revised marginal utility equation, if the vertex x is in close proximity to ν_i, it is conceivable for $\Delta E_{ij}[\zeta_i(t) \oplus \{x\}]$ to be less than or equal to $\Delta E_{ij}[\zeta_i(t)]$, potentially contravening the DMG principle. To mitigate this and ensure convergence when utilizing non-DMG score functions, a warping mechanism is introduced, adjusting the score to $\min_{\xi \subseteq \zeta_i(t)} S_{ij}[\xi]$, thereby aiding the convergence process where traditional DMG is not inherently present.

The score function's direct correlation to both reward and energy consumption raises concerns about its scale invariance. When the mapping of tasks to agents is scaled linearly, the relative value of the scores, and consequently the task allocation decisions, may be altered. In our proposed methodology, we introduce a suite of heuristic extensions to the CBBA, each characterized by a novel scoring function. The diversity of these heuristic extensions is tailored to address the varying demands of distinct allocation problems, potentially outperforming the application of a single heuristic in all scenarios.

The scoring functions for the heuristics are defined as follows:

$$\begin{cases} H_1 & = \gamma_{ij} - \Delta E_{ij}[\zeta_i] \\ H_2 & = \gamma_{ij} \\ H_3 & = \frac{\gamma_{ij}}{\Delta E_{ij}[\zeta_i]} \\ H_4 & = \frac{\gamma_{ij} - \Delta E_{ij}[\zeta_i]}{\mathcal{E}_i[\zeta_i]} \end{cases} \tag{6}$$

where γ_{ij} denotes the reward associated with task j by agent i, $\Delta E_{ij}[\zeta_i]$ represents the additional energy expenditure for incorporating task j into the sequence, and $\mathcal{E}_i[\zeta_i]$ symbolizes the remaining energy budget of agent i.

To optimally leverage these heuristics, we employ a machine learning model trained via a neural network to predict the effectiveness of each heuristic extension. We adopt a Graph Convolutional Network (GCN) tailored for learning from graph-structured data. The architecture of the developed GCN model, as depicted in Figure 2, comprises:

- A tripartite Graph CNN structure with convolutional layers yielding 32-, 16-, and 8-dimensional feature maps, designed to distill environment-specific information such as task connectivity and site distances.
- A mean pooling layer follows to aggregate node features into a comprehensive graph-level representation.
- Two dense layers, each with eight neurons, to process the pooled graph features.
- The model culminates in an output layer that provides a predictive assessment of the aggregate rewards.

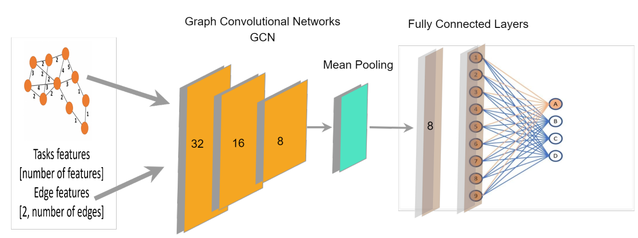

Fig. 2. Proposed Graph Convolutional Network

By integrating the GCN predictions with our heuristic framework, we aim to enhance the decision-making process in the allocation of tasks, ensuring an informed and adaptive approach.

III. Results and Discussion

Figure 3 presents a series of bar charts comparing the performance of the predictive model across four different heuristic methods: H1, H2, H3, and H4. For each heuristic, Series1 represents the values obtained using the heuristic method, while Series2 represents the predicted values generated by the model. The predictions for H1 are closely aligned with the heuristic values, indicating a high degree of accuracy for this method. This is particularly evident in instances where the two series produce almost identical bar heights (e.g., at intervals 1, 4, 6, and 10). This indicates that the model has learned the pattern for H1 and can replicate its decision-making process with high fidelity. For H2, the model appears to have greater variance in its predictive accuracy. While some predictions are quite close to the heuristic values (e.g., intervals 5 and 8), there are others where there is a noticeable difference (e.g., intervals 2 and 9). The model demonstrates a similar pattern of accuracy with the H3 method as with H1, with many of the predictions being close to the heuristic values. The close correspondence in intervals 3, 4, and 7 suggests that the model is largely effective in estimating the H3 heuristic method's outcomes. Lastly, H4 shows a mixed pattern where the model accurately predicts the heuristic values in several intervals (such as 2, 5, and 7), but also deviates significantly in others (such as 1, 8, and 10).

Overall, across all four heuristic methods, the model seems capable of making reasonably accurate predictions and appears to be a promising tool for replicating the patterns in different heuristic methods. However, the variations in predictive accuracy across different methods and intervals show that there are few unique characteristics in each heuristic that the model is variably capturing. Further analysis would be beneficial to understand these differences, refine the model accordingly, and potentially improve its predictive performance.

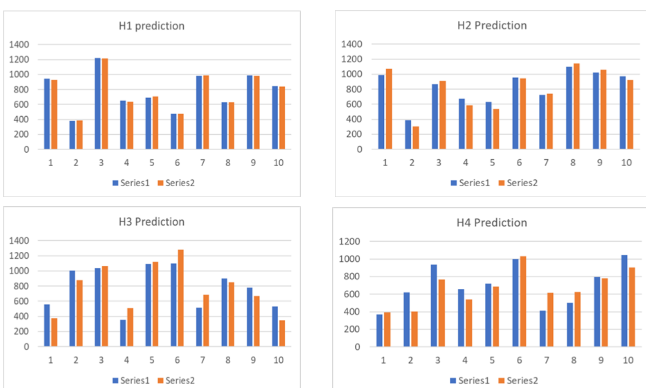

Fig. 3. **Series1: The Heuristic method, Series2: The prediction made by the model**

The two graphs presented in Figure 4 offer a visualization of the performance of AI-enhanced CBBA and Typical CBBA (Original) in comparison to the real score values over a series of tasks or evaluations.It is observed that both AI-CBBA and Typical CBBA closely track the real score values.The AI-enhanced CBBA appears to follow the trend of the real score more tightly than the Typical CBBA, showing that the integration of AI methods within the CBBA framework enhances its ability to mirror the actual score outcomes.

In Figure 5, we present a performance comparison of CBBA, ICBA, Prim's algorithm, and AI-CBBA relative to the number of tasks assigned, with the average distance metric as the evaluation criterion. As the number of tasks increases from 0 to 50, all algorithms exhibit an increasing trend in the average distance, which is intuitive since more tasks typically translate to greater traversal distances for agents. The CBBA shows the steepest increase in distance, indicating that while it

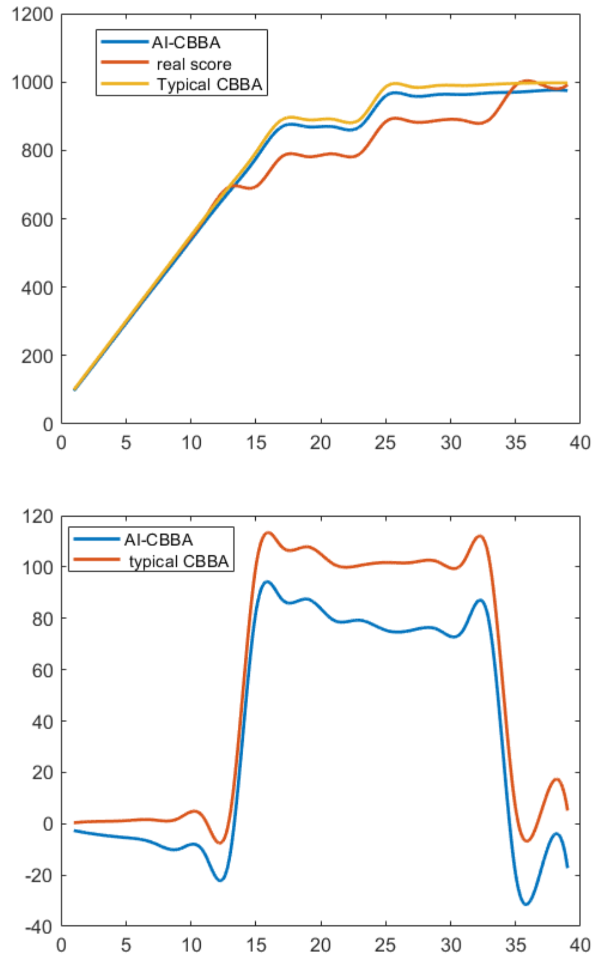

Fig. 4. AI-enhanced CBBA vs Original (Typical) CBBA vs Real score

might be effective for a smaller number of tasks, its efficiency diminishes as the task count rises. The ICBA presents an improvement over CBBA, as evidenced by the lower trajectory of its curve. Prim's algorithm, traditionally used for finding a minimum spanning tree and here adapted for task allocation, demonstrates a performance that initially parallels the ICBA but eventually outperforms it as the number of tasks becomes larger. This indicates Prim's effectiveness in creating more efficient path plans over larger task sets, by leveraging its inherent nature of connecting points in a graph minimally.

Lastly, the AI-CBBA shows the best performance among all the algorithms, maintaining the lowest average distance across the task spectrum. Its curve shows that the incorporation of AI techniques into the standard CBBA framework significantly optimizes the allocation and sequencing of tasks. This optimization stems from the AI's capability to learn from the environment and the ability to predict more efficient allocations or paths.

The data presented in Table II offers insightful revelations concerning the time efficiency of the proposed approach, evaluated under varying operational scales characterized by the

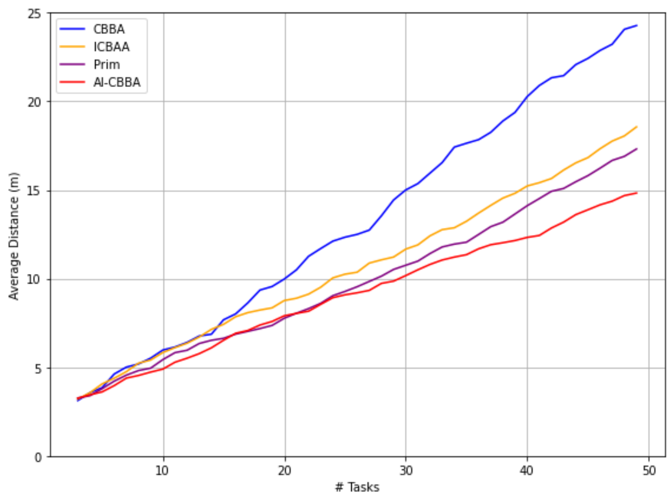

Fig. 5. Average Distance vs Tasks

number of tasks and the agents. The first column, representing the minimum time, consistently increases with the number of agents. The average time, denoted in the second column, similarly ascends with the number of agents. Maximal time in the third column escalates as well with agent count. The standard deviation of time, illustrated in the fourth column, indicates a growing variability in the time to complete tasks.

TABLE II
TIME RESULTS SUMMARY IN [S]

Number of Agents	Min Time	Avg Time	Max Time	Std Dev
2	0.0488	0.5393	0.9020	0.2682
4	0.0917	1.0006	1.5168	0.4583
6	0.1372	1.4613	2.1640	0.6599
8	0.2154	1.9308	2.8104	0.8517
10	0.2462	2.3975	3.4392	1.0542

IV. CONCLUSION

In this study, we successfully integrated Graph Convolutional Networks (GCNs) into the Consensus-Based Bundle Algorithm (CBBA) to enhance task allocation in multi-robot systems, creating an AI-enhanced version (AI-CBBA). This integration marks a shift from traditional heuristic methods to a learning-based approach. AI-CBBA outperforms existing algorithms like original CBBA, Improved CBBA (ICBA), and Prim's algorithm in task allocation efficiency. It excels in managing complex task loads, demonstrating AI's capability to learn and optimize both task allocation and sequencing. Our findings indicate that AI-CBBA could significantly advance multi-robot system coordination, promising improvements for complex operations across various domains.

ACKNOWLEDGMENT

The research presented in this paper was financed by the European Commission and managed by the European Defense Agency in the framework of the Preparatory Action on Defense Research under Grant Agreement 884866 (AIDED).

REFERENCES

[1] M. Campion, P. Ranganathan, and S. Faruque, "Uav swarm communication and control architectures: a review," *Journal of Unmanned Vehicle Systems*, vol. 7, no. 2, pp. 93–106, 2018.

[2] Y. Zhou, B. Rao, and W. Wang, "Uav swarm intelligence: Recent advances and future trends," *Ieee Access*, vol. 8, pp. 183 856–183 878, 2020.

[3] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[4] H.-L. Choi, L. Brunet, and J. P. How, "Consensus-based decentralized auctions for robust task allocation," *IEEE transactions on robotics*, vol. 25, no. 4, pp. 912–926, 2009.

[5] G. A. Korsah, A. Stentz, and M. B. Dias, "A comprehensive taxonomy for multi-robot task allocation," *The International Journal of Robotics Research*, vol. 32, no. 12, pp. 1495–1512, 2013.

[6] L. Johnson, H.-L. Choi, S. Ponda, and J. P. How, "Allowing non-submodular score functions in distributed task allocation," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE, 2012, pp. 4702–4708.

[7] L. B. Johnson, H.-L. Choi, and J. P. How, "The role of information assumptions in decentralized task allocation: A tutorial," *IEEE Control Systems Magazine*, vol. 36, no. 4, pp. 45–58, 2016.

[8] L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 2, pp. 156–172, 2008.

[9] L. Panait and S. Luke, "Cooperative multi-agent learning: The state of the art," *Autonomous agents and multi-agent systems*, vol. 11, pp. 387–434, 2005.

[10] M. Barer, G. Sharon, R. Stern, and A. Felner, "Suboptimal variants of the conflict-based search algorithm for the multi-agent pathfinding problem," in *Proceedings of the International Symposium on Combinatorial Search*, vol. 5, no. 1, 2014, pp. 19–27.

[11] J. P. Van Den Berg and M. H. Overmars, "Prioritized motion planning for multiple robots," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005, pp. 430–435.

[12] T. Standley and R. Korf, "Complete algorithms for cooperative pathfinding problems," in *IJCAI*. Citeseer, 2011, pp. 668–673.

[13] W. Wu, S. Bhattacharya, and A. Prorok, "Multi-robot path deconfliction through prioritization by path prospects," in *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 9809–9815.

[14] C. Luo, Q. Huang, F. Kong, S. Khan, and Q. Qiu, "Applying machine learning in designing distributed auction for multi-agent task allocation with budget constraints," in *2021 20th International Conference on Advanced Robotics (ICAR)*. IEEE, 2021, pp. 356–363.

[15] J. Blumenkamp, S. Morad, J. Gielis, Q. Li, and A. Prorok, "A framework for real-world multi-robot systems running decentralized gnn-based policies," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 8772–8778.

[16] A. Khan, E. Tolstaya, A. Ribeiro, and V. Kumar, "Graph policy gradients for large scale robot control," in *Conference on robot learning*. PMLR, 2020, pp. 823–834.

[17] E. Tolstaya, F. Gama, J. Paulos, G. Pappas, V. Kumar, and A. Ribeiro, "Learning decentralized controllers for robot swarms with graph neural networks," in *Conference on robot learning*. PMLR, 2020, pp. 671–682.

[18] Q. Li, F. Gama, A. Ribeiro, and A. Prorok, "Graph neural networks for decentralized multi-robot path planning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 11 785–11 792.

[19] R. Kortvelesy and A. Prorok, "Modgnn: Expert policy approximation in multi-agent systems with a modular graph neural network architecture," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 9161–9167.

[20] E. Ghisoni, S. Govindaraj, A. M. C. Faulí, G. De Cubber, F. Polisano, N. Aouf, D. Rondao, Z. Chekakta, and B. de Waard, "Multi-agent system and ai for explosive ordnance disposal," *CEIA HUMANITARIAN CLEARANCE TEAMWORK*, p. 26.