

Task Allocation for Autonomous Machines using Computational Intelligence and Deep Reinforcement Learning

Thanh Thi Nguyen¹, Quoc Viet Hung Nguyen², Jonathan Kua³,
Imran Razzak⁴, Dung Nguyen⁵ and Saeid Nahavandi⁶

Abstract—Enabling multiple autonomous machines to perform reliably requires the development of efficient cooperative control algorithms. This paper presents a survey of algorithms that have been developed for controlling and coordinating autonomous machines in complex environments. We especially focus on task allocation methods using computational intelligence (CI) and deep reinforcement learning (RL). The advantages and disadvantages of the surveyed methods are analysed thoroughly. We also propose and discuss in detail various future research directions that shed light on how to improve existing algorithms or create new methods to enhance the employability and performance of autonomous machines in real-world applications. The findings indicate that CI and deep RL methods provide viable approaches to addressing complex task allocation problems in dynamic and uncertain environments. The recent development of deep RL has greatly contributed to the literature on controlling and coordinating autonomous machines, and it has become a growing trend in this area. It is envisaged that this paper will provide researchers and engineers with a comprehensive overview of progress in machine learning research related to autonomous machines. It also highlights underexplored areas, identifies emerging methodologies, and suggests new avenues for exploration in future research within this domain.

I. INTRODUCTION

Autonomous machines are increasingly being used across a wide range of fields, including manufacturing, transportation, agriculture, environmental monitoring, healthcare, retail, mining, and military applications [1]–[3]. A useful autonomous system must be able to operate in various complex scenarios and capable of performing the assigned missions effectively and competently. When the mission complexity increases, it is desirable to share the burden with multiple vehicles or systems. The cooperation of multiple machines can simultaneously increase performance, decrease mission time, and enhance the mission success rate [4], [5].

Several challenges arise from the operation of multiple machines [6], [7]. These challenges are associated directly with *task allocation* (TA) and coordination of systems of unmanned vehicles, which is significantly complicated when the numbers of vehicles is large. The purpose of TA is to assign a series of important and necessary tasks, e.g., reconnaissance, search, attack, and verification, to intelligent

agents so as to maximise the overall mission performance. An effective TA solution is critical for achieving a high degree of efficacy in mission planning. Generally, TA is a combinatorial optimization problem that finds the minimal cost solution between two separate sets: a set of agents $A = \{a_1, a_2, \dots, a_n\}$ and a set of tasks $T = \{t_1, t_2, \dots, t_m\}$. To complete each task, an agent requires a cost. Let us define C_{ij} as the non-negative cost of allocating the i th agent for handling the j th task. The aim is to allocate each task to one agent so as to minimize the total cost of finishing all tasks. The total allocation cost is characterized by:

$$\sum (X_{ij}C_{ij}) \text{ for } i = 1, 2, \dots, n \text{ and } j = 1, 2, \dots, m \quad (1)$$

where X_{ij} is a binary variable with a value of 1 if agent a_i is assigned to task t_j and a value of 0 otherwise. To achieve an efficient allocation, the total cost needs to be minimized, and therefore TA becomes an optimization problem, as follows:

$$\begin{aligned} &\text{find } X_{ij} \\ &\text{subject to } \min \sum (X_{ij}C_{ij}) \\ &\text{for } i = 1, 2, \dots, n \text{ and } j = 1, 2, \dots, m \end{aligned} \quad (2)$$

The existing work in Seenu et al. [8] reviews TA methods for multiple-robot systems, but it does not cover the latest developments in this area, especially the emerging applications of deep reinforcement learning (RL) methods. In this paper, a comprehensive literature review covering computational intelligence and deep RL methods for TA is conducted. Fig. 1 presents the structure of the survey and taxonomy of reviewed techniques. The benefits and drawbacks of the surveyed methodologies are analysed and contrasted thoroughly.

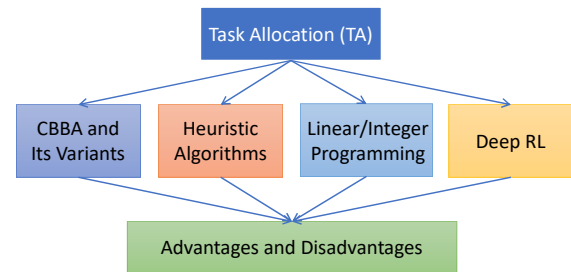


Fig. 1. A taxonomy of TA methods included in this review. CBBA is the abbreviation of consensus-based bundle algorithm while RL stands for reinforcement learning.

¹Faculty of Information Technology, Monash University, Australia
thanh.nguyen9@monash.edu

²Griffith University, Queensland, Australia

³Deakin University, Victoria, Australia

⁴Mohamed bin Zayed University of Artificial Intelligence, UAE

⁵The University of Queensland, Australia

⁶Swinburne University of Technology, Melbourne, Victoria, Australia
snahavandi@swin.edu.au

II. TASK ALLOCATION: A SURVEY

The TA methodologies can be categorised into two groups, i.e., centralised and decentralised methods [9]. In *centralised* TA methods, a single control center obtains the entire information from all agents, and then processes and directs suitable commands to each agent. Linear/integer programming-based techniques are examples of centralized methods that attempt to acquire an optimal but computationally inexpensive solution [10]. With regard to evolutionary approaches, genetic algorithm (GA), which can offer sub-optimal solutions with a reduced computational burden, is a popular method for centralised TA problems [11]. Similarly, particle swarm optimization (PSO) has also been an efficient method for centralized TA problems [12], [13].

In *decentralized* TA methods, each agent acquires local information by itself, or through communications with neighboring agents. Therefore, it can cope with dynamic and unanticipated situations more reliably. In addition, decentralised TA methods can mitigate the computational burden by aggregating the computing power from each constituent agent. Nevertheless, the consensus on situation awareness poses a key challenge in decentralised TA methods. Market-based techniques have been proposed to address this challenge in a decentralized system. Each participant in the virtual market gives a decision based on its own profit and circumstances, therefore increasing the effectiveness of every participant. A market mechanism such as an auction is used to distribute resources to the participants in this virtual market [9].

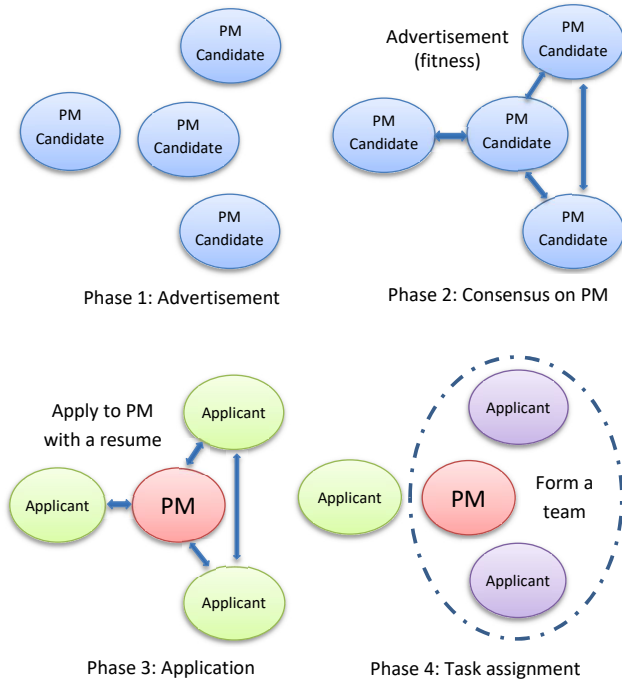


Fig. 2. Four phases of the TA process in a virtual market [9]. The project manager (PM) is the candidate who has the highest fitness score.

Fig. 2 illustrates the concept of the virtual market and its decentralised selection procedure. Agents disclose their

information to others, in order to build a team and to acquire a task, which is known as a project. In the first phase, every agent can advertise regardless of the number of neighbor candidates (Phase 1). The suitability of a candidate to complete a specific task is measured by a fitness function. A candidate is selected as a project manager if it has the highest fitness score (Phase 2). As soon as the manager is determined, other agents send an application resume to the manager including estimated information on completing the applied task (Phase 3). A consensus on the application letters is made among agents by sharing their application letters with their neighborhoods. Using this method, distant agents do not need to use a network topology to reach the manager. Based on the resumes, the manager chooses a number of members among the applicants to form a team (Phase 4). This procedure is repeated to assign a specific task to the team members, and at the end, each agent acquires its own task. A polynomial-time algorithm combining both *auction* and *consensus* methods, known as the consensus-based bundle algorithm (CBBA), was introduced in [14], and detailed in the following subsection.

A. CBBA and its variants

Cooperation in an agent fleet is crucial for improving the overall performance in accomplishing a mission. Centralized methods communicate situation awareness to a centralized control server that produces a plan for the whole fleet. These kinds of systems have the advantage that they put much of the processing burden securely on the ground, allowing the agents (machines) to be smaller and more cost-effective to build. However, agents need to constantly communicate with the fixed server, therefore decreasing the mission handling power of the fleet, as well as generating a single point of control, which is vulnerable to failure in the system.

To increase the ability to handle a large mission range and also to eliminate the single point of failure, decentralized methods equip each agent with a centralized planner. In case of limited communication, which is not uncommon in realistic networks, inconsistencies in situation awareness might lead to conflicting TA because agents obtain optimal solutions based on different sets of information. As a result, *consensus algorithms* are often used to formulate a consistent situation awareness before TA takes place. These consensus algorithms make sure that situation awareness is converged over different network topologies, which allows the system to perform TA in highly dynamic and uncertain situations. The consensus algorithms have disadvantages as they take a considerable amount of time to converge to a consistent situation awareness condition, and often require transmitting a large amount of data. This can lead to severe delays in low-bandwidth environments, and considerably increase the total TA time for the fleet.

Another class of methods for TA is *auction algorithms*. Generally, agents place bids on tasks with values using their own situation awareness, and the greatest bid wins the allocation. Each task is to be allocated to a single agent as only one agent is chosen as the winner. Therefore,

convergence to a conflict-free solution can be reached for most auction algorithms even agents may have inconsistent situation awareness. The shortcoming of these algorithms is that the bids from the agents must be transferred to the auctioneer. The types of network topologies are therefore limited because a connected network is needed among the agents to convey all the bid information.

Consensus methods are generally robust to network topologies, whereas traditional auction methods are computationally effective and robust to inconsistencies in situation awareness. To take their advantages, a combination of these two methods, namely CBBA, was introduced in [14].

The consensus-based grouping algorithm (CBGA) proposed in [15], [16] extends CBBA by introducing a new data storage system and a multiagent consensus algorithm to handle tasks that need more than one agent to complete. CBGA can handle multiagent multi-TA with multiple agent requirements pertaining to UAVs. Conventional CBBA assumes homogeneity of agents, which is often violated in real practice. Heterogeneous robots consensus-based allocation (HRCA) suggested in [17] extends CBBA by handling constraint violations in the second stage. The agents (robots) use an iterative procedure to re-distribute the tasks that exceed their individual capability with a minimal loss in terms of the score function. Therefore, HRCA is able to handle randomly generated heterogeneous robotic networks [17].

B. Heuristic algorithms for TA problems

Evolutionary algorithms are generic metaheuristic techniques that perform optimization based on a population of evolutionary individuals. They employ mechanisms inspired by biological evolutionary processes to discover optimal or near-optimal solutions for complex and sometimes NP-hard problems. These mechanisms include reproduction, mutation, recombination, and selection. One of the key evolutionary algorithms is GA, which was utilized in numerous applications [18]–[20]. Other heuristics that have been broadly employed for TA include swarm-based algorithms, such as PSO.

GA has a number of advantages for undertaking TA problems. It does not require explicit calculation of the gradient of the cost function and can handle nonlinear cost functions. GA is able to produce solutions that are significantly better than those from the random search method and it has a possibility of parallel implementation [11].

On the one hand, PSO [21]–[23] runs faster than GA in TA. This is because PSO attempts to balance exploration (through neighboring experience) and exploitation (through self-experience) by combining global and local search methods, respectively. As such, it has a great ability to escape from local optima. On the other hand, a variant of PSO, i.e., discrete PSO [21], performs better than PSO, GA, simulated annealing, and tabu search in terms of search quality to expedite convergence by embedding a variable neighborhood descent search method based on a transfer and swap neighborhood. Good performance of discrete PSO also results from a new migration mechanism to strike an equilibrium between extensive exploration and targeted exploitation [21].

In case of large, self-operating, multi-robot systems, scalability and the capability of adapting to different environments are the utmost importance feature of TA algorithms. Based on the bee algorithms [24], [25], a distributed bee algorithm was proposed in [26], [27] for assigning robots to different positions in a 2-D arena. The qualities of targets, which are characterized by scalar values, are used to derive the expected distribution. The decision-making system is distributed, and the distances and qualities of targets are taken into account so that the robots can autonomously obtain their allocated tasks. The distributed bee algorithm provides a robot swarm with not only the flexibility where the total of robots and targets can be varied but also the ability to adapt to a non-uniform distribution of the attributes of targets [27].

Many studies have focused on the TA problem of homogeneous UAVs, but the task diversity requires a higher capability of a UAV system. A modified GA for TA was proposed in [28] for a set of heterogeneous UAVs with limited resource constraints. They introduced the idea of a fuzzy elite degree to enhance gene retention and intensity in GA operations. For example, in the GA selection process, the fuzzy elite degree of the i th individual is defined by:

$$E_i = \alpha F_i + \beta \Delta_i \quad (3)$$

where α and β are coefficients with $\alpha + \beta = 1$. These coefficients respectively control the influence of F_i and Δ_i with their definitions as follows:

$$F_i = \frac{\log_2(f_i / \max_i(f_i) + 1)}{\max_i(\log_2(f_i / \max_i(f_i) + 1))} \quad (4)$$

and

$$\Delta_i = \frac{\max_i(1 + \exp(10 * (A_i / \max_i(A_i) - l_s)))}{1 + \exp(10 * (A_i / \max_i(A_i) - l_s))} \quad (5)$$

where f_i and A_i are the fitness score and the number of times the i th individual appears in the current iteration, and l_s is used to adjust the range of A_i . Accordingly, individuals having higher elite degrees will be chosen as elite ones. It means that individuals with high fitness value f are selected with high probability, but individuals with few A also have a chance to be chosen to maintain diversity within the population. Experimental results on a complex TA problem of a heterogeneous UAV system demonstrate an improved searchability and convergence performance of this approach compared with its competing methods.

C. Integer/linear programming (LP) for TA

While CBBA (and its variants) and heuristic approaches have shown remarkable performance in complex TA problems, the use of classical integer/LP methods is still growing. For example, a mission planning problem for military aircraft was considered in [29] in which a given fleet of aircraft attacks several ground targets. The airspace is discretised such that the strike positions and lighting positions can be characterized by nodes. Aircraft movements are represented by arcs, which are defined by a priori knowledge. The goal is to optimize the outcome and mission timespan of the attacks. The problem is modeled as an expanded vehicle routing

problem where timing and precedence constraints are taken into account. A mixed-integer linear programming is then used to solve the problem.

In numerous practical situations, robot fleets must address not only delivery requests but also additional factors like energy efficiency and the need to avoid human-centered workspaces (e.g., avoiding areas with high foot traffic). The study in [30] framed the multi-robot pickup and delivery problem as a multiobjective optimization challenge, aiming to identify policies that balance various tradeoffs among these objectives. A key aspect of that problem is accounting for the variability in objective values due to the unpredictable nature of task arrivals. That study introduces an adaptive sampling method designed to identify policies that are nearly optimal, approximate the full range of most effective solutions, and are statistically different from one another. The method demonstrates clear advantages over baselines and shows robustness through sensitivity analysis.

TA and safety are two critical challenges in collaborative robotics because they involve the interaction between two key resources, the operator and the collaborative robot (cobot), to achieve optimal performance while maintaining a safe workspace. However, most methods tend to prioritize one of these issues over the other. To address this gap, a model based on integer programming was proposed in [31] for collaborative assembly systems, incorporating safety constraints with the goal of minimizing the total task completion time. The innovation of this model lies in its consideration of the distance between resources and its ability to adjust the cobot's speed accordingly. By allowing the cobot to maintain higher speeds for extended periods, the model ensures that the safety constraint is met by maximizing the time resources remain at a safe distance from each other. This approach effectively balances performance and safety in collaborative settings. Although experimental results validate and underscore the necessity of the proposed method, one of its limitations is that it does not account for stochastic times or precedence constraints, which could restrict the range of possible solutions.

D. Deep Reinforcement Learning (RL) for TA

1) *Deep RL Background:* RL is a class of machine learning methods, which are the backbone of many sequential decision-making support systems. RL tasks are normally characterised by a Markov decision process (MDP), which establishes a mathematical framework for analyzing sequential decision-making problems [32]. An MDP is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ where $\mathcal{S}$ is a collection of states, $\mathcal{A}$ is a collection of actions, $\mathcal{P}$ represents a matrix of state transition probabilities $\mathcal{P}_{ss'} = \mathbb{P}[S_{t+1} = s' | S_t = s, A_t = a]$, $\mathcal{R}$ signifies a reward function $\mathcal{R}_s = \mathbb{E}[R_{t+1} | S_t = s, A_t = a]$, and γ is a discount factor $\gamma \in [0, 1]$. The behaviors of an agent can be defined by a policy π , which is the likelihood of taking an action given a state:

$$\pi(a|s) = \mathbb{P}[A_t = a | S_t = s] \quad (6)$$

To find an optimal policy, we strive to optimize the

expected return, $\mathbb{E}[G_t]$, at each step t , where G_t is the total discounted reward:

$$\begin{aligned} G_t &= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots \\ &= \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \end{aligned} \quad (7)$$

Optimal policies, denoted as π_* , are the ones that are superior to or at least equivalent to all the others. These optimal policies feature similar optimal state-value $v_*(s) = \max_{\pi} v_{\pi}(s)$ for all $s \in \mathcal{S}$ where $v_{\pi}(s)$ is the state-value function of an MDP, which is the estimated return by beginning with state s , and then pursuing policy π :

$$\begin{aligned} v_{\pi}(s) &= \mathbb{E}_{\pi}[G_t | S_t = s] \\ &= \mathbb{E}_{\pi}[R_{t+1} + \gamma v_{\pi}(S_{t+1}) | S_t = s] \end{aligned} \quad (8)$$

Complex problems normally have a large number of states, which hinder algorithms such as dynamic programming to find direct solutions. Approximate solutions by RL methods are therefore advocated. For high-dimensional problems where state and action spaces are large, conventional RL methods may struggle to learn optimal policies effectively. Deep learning, known for its ability to solve high-dimensional problems, has been integrated into RL methods to form what is called deep RL. Deep RL therefore becomes an excellent tool for applications where decisions must be made in large-scale environments. Deep RL supports ongoing learning and can create agents that are more reliable and scalable than their human counterparts [33].

2) *Applications of Deep RL for TA:* In a multiagent system, an agent is normally unable to obtain information or observe the behaviors of other agents. To perform complex tasks cooperatively, agents can learn to communicate via a communication channel before taking actions. A TA method based on deep RL was introduced in [34], namely cooperative deep Q-learning, to improve the communication and social cooperation between agents based on the communication neural net [35]. Three types of agents are defined, including manager, participant, and mediator. The manager is an agent that can request help from other agents to perform its task while the participant is an agent that accepts and carries out the declared task. The mediator is an agent that commits to find participants for another agent. An agent can be in one of the three states, i.e., busy, committed, or idle, at any time step. The manager or participant agent will have the busy state while the mediator agent will have the committed state. An agent that is free and not allocated or committed to any task will have the idle state and only this kind of agent can be allocated a new task.

Once the complex problem is broken down into several tasks, the manager agent will send messages to all of its neighbors and idle agents will submit bids to perform the tasks. Depending on the resources of these neighbors, the manager will form a full group or a partial group for the announced task. In case of a partial group, the manager can commit an unfulfilled task to a mediator agent, which is one of the neighboring agents. An agent is more likely

to be selected as a mediator if it has more neighbors. This is a repeated process until the resources required for the announced task are met. Experimental results by using different types of social networks as in [36] demonstrate a great performance of the proposed cooperative deep RL approach. The agents are able to learn not only to allocate tasks efficiently but also to prepare properly for new tasks.

The development of wireless technology, e.g., the 5G network, has enabled the continuing enhancement of the Internet of Vehicles, and that has attracted more and more users. Mobile edge computing (MEC) has provided the edge server infrastructure to facilitate the data processing and transmission tasks to improve the quality of experience of vehicle users. Given the increasing demand for multimedia services from vehicle users and the fast-moving speed of vehicles, it is still challenging to ensure efficient and stable data transmission to users. A pre-caching and TA approach based on deep RL was proposed in [37] to address this problem in a simulation environment consisting of multiple roadside units and vehicles. The RL algorithm is used to learn an intelligent policy to cache data and allocate the data transmission tasks collaboratively between the vehicle to roadside unit and vehicle to vehicle communication modes. Simulation results exhibit the efficiency of the RL method and its superiority against competing policies in terms of maximizing the data receiving rate of fast-moving vehicles.

With the low-cost and strong mobility capabilities, UAVs have been used to improve the connectivity in the MEC technology. A traditional MEC normally has edge servers in fixed cellular base stations, which can negatively affect the service quality provided to mobile users. Multiple UAVs can be used as mobile servers in the MEC to enhance the computational offloading services to mobile devices. However, it is challenging to solve the TA problem for UAVs in this kind of UAV-assisted MEC system because of its non-convex characteristics. Several studies have attempted to solve this problem using greedy algorithm [38], Lyapunov optimization method [39], Q-learning [40], and iterative algorithm [41], but they have limitations in handling action spaces that are high-dimensional and continuous. On the other hand, this problem was formulated in [42] as an MDP and used the twin delayed deep deterministic policy gradient proposed in [43] to find optimal solutions in a continuous action space. Each UAV is characterized by a deep RL agent implemented by the twin delayed deep deterministic RL algorithm [44].

On the other hand, a multiagent RL method was designed in [45] to dynamically schedule tasks in cooperative multi-robot settings, considering the service status of industrial robots. The approach establishes constraints that define the interactions between tasks and robots and develops serviceability evaluation and error compensation agents within the deep deterministic policy gradient RL framework to evaluate robot serviceability and correct positioning errors according to their operational status. Furthermore, a heuristic graph convolution scheduling agent is created by integrating heuristic rules with graph CNNs to assign tasks according to the characteristics of both tasks and robots and their in-

terdependencies. The approach facilitates both static TA and adaptive scheduling in multi-robot collaborations, effectively addressing the single-task, multi-robot, and time-extended assignment problem in industrial settings.

Design and planning for a multi-robot system often involve two basic steps, i.e., TA and task execution. Most approaches are for either TA or path planning. An end-to-end deep RL approach to both multi-robot TA and path planning was proposed in [46] based on the deep RL proximal policy optimization method [47]. The experiments are performed on a 2D plane environment having both fixed and moving obstacles. The robots are homogeneous and the number of robots is equal to the number of goal locations, i.e., targets. Each robot is encouraged to select and approach a specific target and aim to minimize its moving time to the target. The robot is penalized if it collides with obstacles or other robots or reaches the same target as other robots. The TurtleBot 3 Waffle Pi robots [48] are used for experiments with the training and testing based on the Gazebo simulator. The results show better performance regarding the success rate, moving time, and distance of the proposed method compared with methods focused solely on navigation.

III. PROS AND CONS OF TA METHODS

The advantages of consensus-based methods, i.e., CBBA and its variants, stem from the integration of market-based auction strategy and consensus method [49]–[52]. The auction approach is used as a framework for decentralized TA while the consensus method based on local communication is applied to resolve conflicts and to reach an agreement on the winning bid. Therefore, CBBA and its variants have better performance and faster convergence speed as compared with the established auction-based TA methods [14].

CBBA and its variants however have a number of drawbacks as pointed out in the literature when applied to complex TA problems [14]–[16], [53], [54]. CBBA does not facilitate the assignment of tasks with precedence requirements [54]. CBBA requires a multiagent system to know the global presence of a task. When the agents search their areas of responsibility and find new targets, the presence of these new targets or tasks must be shared with each agent, and the procedure needs to be re-run with the newly acquired information [54].

More importantly, CBBA can only give conflict-free solutions with a guaranteed 50% optimality when solving multiagent multi-TA problems [14]–[16]. It was highlighted in [53] that CBBA does not explicitly possess the ability to address obstacle avoidance and its allocation results may be highly susceptible to input noise and can produce churning behaviors. Alternatively, CBBA can only handle homogeneous robotic systems in which robots are assumed to have similar characteristics and capabilities as pointed out in [17].

The extensions of CBBA, e.g., extended CBBA [54] or CBGA [15], [16] have demonstrated that they are able to overcome the shortcomings of CBBA, and can be applied to complex scenarios. HRCA can be applied for TA of UAVs where the networks of UAVs are heterogeneous, i.e., UAVs

are not identical owing to their sensing or actuation hardware and features [17].

With regard to the evolutionary algorithms, they have been found to be increasingly used for TA. GA is one of several traditional methods in this category. Although GA has been widely used for addressing TA problems [11], it has a significant disadvantage related to its computational expense [22]. GA also generates solutions with higher costs as compared to PSO [22]. Extensions of PSO, e.g., discrete PSO [21], hybrid PSO [22], and reverse predictor PSO [13] show efficiency in terms of convergence speed and average costs. Other evolutionary algorithms developed for TA of robot swarms, e.g., distributed bee algorithm [27] or processors in distributed systems, e.g., novel global harmony search algorithm [55], have a good potential for applications to complex TA problems.

On the other hand, deep RL methods have attracted huge attention recently and their applications for TA have emerged. They are an end-to-end approach as they can take data directly from the sensors and learn a policy that is able to generate optimal sequential solutions for autonomous machines. Deep RL methods have a disadvantage that their training process may require significant computational resources because building a deep RL agent is only effective when it has a large amount of environment interactions [56]. Furthermore, the input sensory data for deep RL algorithms are normally obtained in terms of high-dimensional images/signals of the environment. The training time of deep RL methods therefore may be greater than heuristics and evolutionary computation algorithms because of the time needed to process high-dimensional data and allow the RL agent to sufficiently engage with the environment. Taking the input environment via its sensory data representation however may be an advantage of deep RL methods because this kind of modeling allows more realistic features to be taken into account [57]. With the power of representation learning, deep neural networks can learn and obtain a compact feature set from raw high-dimensional images or sensory data of the environment, which enables the network to approximate complex functions describing the relationship between inputs and action outputs. Deep RL methods therefore have made significant contributions in control and coordination of autonomous machines as reviewed in this study.

IV. OPEN ISSUES AND FUTURE DIRECTIONS

A. Benchmarking TA methods

There have been numerous TA methods reviewed in this study, but they are mostly evaluated using different environments and experiment settings. In each category, there are traditional methods with their improved versions still being developed over time, but there are also new approaches that benefit from recent machine learning advances such as deep RL. Each method has its own merits and demerits depending on its nature and the environment in which it is used. For example, a linear programming method can be very efficient in a simple TA problem, but its performance may not be comparable with that of an evolutionary computation method

in a more sophisticated problem. In the TA literature, there is currently no comprehensive work on comparing variants of CBBA with linear/integer programming, GA, PSO, bat algorithm, bee algorithm, harmony search, and the emerging deep RL methods using a benchmark environment setting. Benchmarking different TA methods in an empirical study is therefore highly recommended as it would provide a comparative evaluation of these methods and give insights into the employability of each method in specific environments. This is a research direction that would promote broader applications of the existing TA methods and also facilitate the development of new and effective methods in this area.

B. Enabling heterogeneous machines

Most approaches proposed for TA have been dealing with machines that have the same or similar features and functions, i.e., homogeneous agents. Real-world problems normally have to cope with complex tasks requiring agents to be heterogeneous machines with different features and functions [58]. For example, UAVs operate in the air while robots perform their tasks on the ground. The tasks assigned for UAVs and robots therefore are dissimilar, and thus the existing methods may struggle to handle them. Recent studies have attempted to address this research topic, but they are still not to the full extent. For example, a multi-robot task and motion planning in a manufacturing environment was introduced in [59], but their experiments are limited to two types of robots that have similar characteristics. Further studies in this direction are important to accommodate more variety of machines and thus maximize the applicability of autonomous machines to real-world problems.

C. The IoT to facilitate interactions between machines

In an environment with multiple autonomous machines, the interactions among the machines are important for efficient route planning and TA. The IoT technologies can be utilized to facilitate these interactions as IoT devices can be mounted on the machines and various static and dynamic objects. The IoT can enable autonomous machines to interact with their environment, recognize obstacles, and move through them. This would create a well-connected, dynamic, and ubiquitous environment with rich data sources including the position of objects, the existence of obstacles, and the changes occurring in the environment [60]. While TA in such environments is more challenging, it also presents significant opportunities to enhance the performance of these algorithms. A future research in this direction is to incorporate the full extent of IoT into TA. IoT-enabled machines will be an emerging research trend in the coming years.

V. CONCLUSIONS

It is difficult for humans to perform mission planning with large systems of machines. Artificial intelligence and machine learning have been applied to automate this process. Collaboration and sharing of information among vehicles are required to maximise the effectiveness when performing missions involving multiple vehicles. In this paper, we have

TABLE I
A SUMMARY OF REVIEWED METHODS FOR TA

Algorithms	Surveyed Methods
Linear/integer programming	[29]–[31], [61]–[63]
CBBA and its variants	[9], [14]–[17], [49]–[54]
Genetic algorithm	[11], [28], [64]–[66]
PSO, ant colony, swarm intelligence	[12], [13], [21], [23], [67]–[69]
Bees/bat algorithm, harmony search	[24]–[27], [55], [70], [71]
Deep reinforcement learning	[37], [42], [45], [46], [72]–[74]

presented a literature review of computational intelligence and deep RL methodologies for TA in changing environments. A summary of the reviewed methods is presented in Table I. Each class of methods however has its respective advantages and disadvantages, as discussed in this paper. While computational intelligence and deep RL methods constitute a great direction to tackle TA problems for autonomous machines, further in-depth research is required to properly engage the related methodologies. We have suggested several interesting research directions, which are worth investigating as they will certainly contribute to improving the efficiency and effectiveness of machine learning and artificial intelligence methods for controlling and coordinating autonomous machines in complex and dynamic real-world problems.

REFERENCES

- [1] L. Huang, M. Zhou, H. Han, S. Wang, and A. Albeshri, "Learning-inspired immune algorithm for multiobjective-optimized multirobot maritime patrolling," *IEEE Internet of Things Journal*, vol. 11, no. 6, pp. 9870–9881, 2023.
- [2] L. Crosato, K. Tian, H. P. Shum, E. S. Ho, Y. Wang, and C. Wei, "Social interaction-aware dynamical models and decision-making for autonomous vehicles," *Advanced Intelligent Systems*, vol. 6, no. 3, p. 2300575, 2024.
- [3] Z. Zhang, Z. Zhao, S. Qin, S. Liu, and M. Zhou, "Multi-product multi-stage multi-period resource allocation for minimizing batch-processing steel production cost," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 5272–5283, 2024.
- [4] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, "Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications," *IEEE Transactions on Cybernetics*, vol. 50, no. 9, pp. 3826–3839, 2020.
- [5] Y. Li, Q. Wang, X. Li, L. Gao, L. Fu, Y. Yu, and W. Zhou, "Real-time scheduling for flexible job shop with AGVs using multiagent reinforcement learning and efficient action decoding," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 55, no. 3, pp. 2120–2132, 2025.
- [6] N. D. Nguyen, T. Nguyen, and S. Nahavandi, "Multi-agent behavioral control system using deep reinforcement learning," *Neurocomputing*, vol. 359, pp. 58–68, 2019.
- [7] A. Miele, M. Lippi, and A. Gasparri, "A distributed framework for integrated task allocation and safe coordination in networked multi-robot systems," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 11 219–11 238, 2025.
- [8] N. Seenu, K. C. RM, M. Ramya, and M. N. Janardhanan, "Review on state-of-the-art dynamic task allocation strategies for multiple-robot systems," *Industrial Robot*, vol. 47, pp. 929–942, 2020.
- [9] G. Oh, Y. Kim, J. Ahn, and H.-L. Choi, "Market-based task assignment for cooperative timing missions over networks with limited connectivity," in *AIAA Guidance, Navigation, and Control Conference*, 2015, doi: 10.2514/6.2015-2005.
- [10] M. Radmanesh and M. Kumar, "Flight formation of UAVs in presence of moving obstacles using fast-dynamic mixed integer linear programming," *Aerospace Science and Technology*, vol. 50, pp. 149–160, 2016.
- [11] M. Darrah, E. Fuller, T. Munasinghe, K. Duling, M. Gautam, and M. Wathen, "Using genetic algorithms for tasking teams of raven UAVs," *Journal of Intelligent & Robotic Systems*, vol. 70, no. 1, pp. 361–371, 2013.
- [12] X. Z. Wang and J. B. Xiao, "Applied research of ADSHPSO algorithm in multi-UAV cooperative mission planning," in *Applied Mechanics and Materials*, vol. 373. Trans Tech Publ, 2013, pp. 1106–1109.
- [13] L. Li, F. Liu, G. Long, P. Guo, and X. Bie, "Modified particle swarm optimization for BMDS interceptor resource planning," *Applied Intelligence*, vol. 44, no. 3, pp. 471–488, 2016.
- [14] 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.
- [15] S. Hunt, Q. Meng, and C. J. Hinde, "An extension of the consensus-based bundle algorithm for multi-agent tasks with task based requirements," in *11th International Conference on Machine Learning and Applications*, vol. 2. IEEE, 2012, pp. 451–456.
- [16] S. Hunt, Q. Meng, C. Hinde, and T. Huang, "A consensus-based grouping algorithm for multi-agent cooperative task allocation with complex requirements," *Cognitive Computation*, vol. 6, no. 3, pp. 338–350, 2014.
- [17] D. Di Paola, D. Naso, and B. Turchiano, "Consensus-based robust decentralized task assignment for heterogeneous robot networks," in *American Control Conference*. IEEE, 2011, pp. 4711–4716.
- [18] A. Khatami, P. M. Kebria, S. M. J. Jalali, A. Khosravi, A. Nazari, M. Shamszadeh, T. Nguyen, and S. Nahavandi, "A GA-based pruning fully connected network for tuned connections in deep networks," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2019, pp. 3492–3497.
- [19] S. M. A. Tousi, S. Samizadeh, and A. Nikoofard, "Genetic algorithm in traffic control and autonomous driving," in *Frontiers in Genetics Algorithm Theory and Applications*. Springer, 2024, pp. 195–208.
- [20] L. Klampfl, F. Klück, and F. Wotawa, "Using genetic algorithms for automating automated lane-keeping system testing," *Journal of Software: Evolution and Process*, vol. 36, no. 3, p. e2520, 2024.
- [21] Q. Kang and H. He, "A novel discrete particle swarm optimization algorithm for meta-task assignment in heterogeneous computing systems," *Microprocessors and Microsystems*, vol. 35, pp. 10–17, 2011.
- [22] P.-Y. Yin, S.-S. Yu, P.-P. Wang, and Y.-T. Wang, "A hybrid particle swarm optimization algorithm for optimal task assignment in distributed systems," *Computer Standards & Interfaces*, vol. 28, no. 4, pp. 441–450, 2006.
- [23] L.-y. Li, F. Liu, and Y. Mei, "Attractor particle swarm optimization for anti-TBM firepower-target match modeling in terminal phase," *Systems Engineering and Electronics*, vol. 35, no. 5, pp. 993–999, 2013.
- [24] L. Özbakir, A. Baykasoğlu, and P. Tapkan, "Bees algorithm for generalized assignment problem," *Applied Mathematics and Computation*, vol. 215, no. 11, pp. 3782–3795, 2010.
- [25] P. Tapkan, L. Özbakir, and A. Baykasoğlu, "Solving fuzzy multiple objective generalized assignment problems directly via bees algorithm and fuzzy ranking," *Expert Systems with Applications*, vol. 40, no. 3, pp. 892–898, 2013.
- [26] A. Jevtić and A. Gutiérrez, "Distributed bees algorithm parameters optimization for a cost efficient target allocation in swarms of robots," *Sensors*, vol. 11, no. 11, pp. 10 880–10 893, 2011.
- [27] A. Jevtic, A. Gutiérrez, D. Andina, and M. Jamshidi, "Distributed bees algorithm for task allocation in swarm of robots," *IEEE Systems Journal*, vol. 6, no. 2, pp. 296–304, 2012.
- [28] S. Han, C. Fan, X. Li, X. Luo, and Z. Liu, "A modified genetic algorithm for task assignment of heterogeneous unmanned aerial vehicle system," *Measurement and Control*, vol. 54, no. 5–6, pp. 994–1014, 2021.
- [29] N.-H. Quttineh, T. Larsson, K. Lundberg, and K. Holmberg, "Military aircraft mission planning: a generalized vehicle routing model with synchronization and precedence," *EURO Journal on Transportation and Logistics*, vol. 2, no. 1–2, pp. 109–127, 2013.
- [30] N. Wilde and J. Alonso-Mora, "Statistically distinct plans for multi-objective task assignment," *IEEE Transactions on Robotics*, vol. 40, pp. 2217–2232, 2024.
- [31] M. Faccio, I. Granata, and R. Minto, "Task allocation model for human-robot collaboration with variable cobot speed," *Journal of Intelligent Manufacturing*, vol. 35, no. 2, pp. 793–806, 2024.
- [32] D. Goel, K. Moore, J. Wang, M. Kim, and T. T. Nguyen, "Unveiling the black box: A multi-layer framework for explaining reinforcement learning-based cyber agents," *arXiv preprint arXiv:2505.11708*, 2025.

- [33] N. D. Nguyen, T. Nguyen, S. Nahavandi, A. Bhatti, and G. Guest, "Manipulating soft tissues by deep reinforcement learning for autonomous robotic surgery," in *2019 IEEE International Systems Conference (SysCon)*. IEEE, 2019, pp. 1–7.
- [34] D. B. Noureddine, A. Gharbi, and S. B. Ahmed, "Multi-agent deep reinforcement learning for task allocation in dynamic environment," in *The 12th Intl. Conf. on Software Technologies*, 2017, pp. 17–26.
- [35] S. Sukhbaatar, R. Fergus *et al.*, "Learning multiagent communication with backpropagation," *Advances in Neural Information Processing Systems*, vol. 29, pp. 2244–2252, 2016.
- [36] M. De Weerd, Y. Zhang, and T. Klos, "Distributed task allocation in social networks," in *The 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, 2007, pp. 1–8.
- [37] T. Ma, X. Chen, Z. Ma, and Y. Chen, "Deep reinforcement learning for pre-caching and task allocation in internet of vehicles," in *IEEE Intl. Conf. on Smart Internet of Things (SmartIoT)*, 2020, pp. 79–85.
- [38] Y. Wang, Z.-Y. Ru, K. Wang, and P.-Q. Huang, "Joint deployment and task scheduling optimization for large-scale mobile users in multi-UAV-enabled mobile edge computing," *IEEE Transactions on Cybernetics*, vol. 50, no. 9, pp. 3984–3997, 2019.
- [39] G. Wang, X. Yu, F. Xu, and J. Cai, "Task offloading and resource allocation for UAV-assisted mobile edge computing with imperfect channel estimation over Rician fading channels," *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, no. 1, pp. 1–19, 2020.
- [40] I. A. Elgendy, W.-Z. Zhang, H. He, B. B. Gupta, and A. A. Abd El-Latif, "Joint computation offloading and task caching for multi-user and multi-task MEC systems: reinforcement learning-based algorithms," *Wireless Networks*, vol. 27, no. 3, pp. 2023–2038, 2021.
- [41] Y. He, D. Zhai, F. Huang, D. Wang, X. Tang, and R. Zhang, "Joint task offloading, resource allocation, and security assurance for mobile edge computing-enabled UAV-assisted VANETs," *Remote Sensing*, vol. 13, no. 8, p. 1547, 2021.
- [42] C. Yu, W. Du, F. Ren, and N. Zhao, "Deep reinforcement learning for task allocation in UAV-enabled mobile edge computing," in *Intl. Conf. on Intell. Networking and Collaborative Sys.*, 2021, pp. 225–232.
- [43] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International Conference on Machine Learning*. PMLR, 2018, pp. 1587–1596.
- [44] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *ICLR*, 2016, <http://arxiv.org/abs/1509.02971>.
- [45] J. Zhou, L. Zheng, and W. Fan, "Multirobot collaborative task dynamic scheduling based on multiagent reinforcement learning with heuristic graph convolution considering robot service performance," *Journal of Manufacturing Systems*, vol. 72, pp. 122–141, 2024.
- [46] A. Elfakharany and Z. H. Ismail, "End-to-end deep reinforcement learning for decentralized task allocation and navigation for a multi-robot system," *Applied Sciences*, vol. 11, no. 7, p. 2895, 2021.
- [47] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv:1707.06347*, 2017.
- [48] A. Kim, D. Seon, D. Lim, H. Cho, J. Jin, L. Jung, M. Y. W. Son, and Y. Pyo, "ROBOTIS TurtleBot3 e-Manual," <https://emanual.robotis.com/docs/en/platform/turtlebot3/overview/>, Accessed: March 2025.
- [49] L. Brunet, H.-L. Choi, and J. How, "Consensus-based auction approaches for decentralized task assignment," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2008, p. 6839.
- [50] M. Argyle, D. W. Casbeer, and R. Beard, "A multi-team extension of the consensus-based bundle algorithm," in *Proceedings of the 2011 American Control Conference*. IEEE, 2011, pp. 5376–5381.
- [51] B. Mu, G. Chowdhary, and J. P. How, "Value-of-information aware active task assignment," in *Algorithms for Synthetic Aperture Radar Imagery XX*, vol. 8746, 2013, p. 87460P.
- [52] G. Oh, Y. Kim, J. Ahn, and H.-L. Choi, "Market-based distributed task assignment for rendezvous mission over networks with limited connectivity," *IFAC Proceedings*, vol. 47, no. 3, pp. 888–893, 2014.
- [53] L. Bertuccelli, H.-L. Choi, P. Cho, and J. How, "Real-time multi-UAV task assignment in dynamic and uncertain environments," in *AIAA Guidance, Navigation, and Control Conference*, 2009, p. 5776.
- [54] T. Mercker, D. W. Casbeer, P. T. Millett, and M. R. Akella, "An extension of consensus-based auction algorithms for decentralized, time-constrained task assignment," in *Proceedings of the American Control Conference*. IEEE, 2010, pp. 6324–6329.
- [55] D. Zou, L. Gao, S. Li, J. Wu, and X. Wang, "A novel global harmony search algorithm for task assignment problem," *Journal of Systems and Software*, vol. 83, no. 10, pp. 1678–1688, 2010.
- [56] T. T. Nguyen, C. M. Nguyen, T. Huynh-The, Q.-V. Pham, Q. V. H. Nguyen, I. Razzak, and V. J. Reddi, "Solving complex sequential decision-making problems by deep reinforcement learning with heuristic rules," in *International Conference on Computational Science*. Springer, 2023, pp. 298–305.
- [57] T. Nguyen, N. D. Nguyen, F. Bello, and S. Nahavandi, "A new tensioning method using deep reinforcement learning for surgical pattern cutting," in *2019 IEEE International Conference on Industrial Technology (ICIT)*. IEEE, 2019, pp. 1339–1344.
- [58] Y. Li and H. Huang, "Efficient task planning for heterogeneous AGVs in warehouses," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 8, pp. 10005–10019, 2024.
- [59] H. Zheng and Y. Wang, "Parallel decomposition and concurrent satisfaction for heterogeneous multi-robot task and motion planning under temporal logic specifications," *Discrete Event Dynamic Systems*, pp. 1–36, 2021.
- [60] Y. Tashtoush, I. Haj-Mahmoud, O. Darwish, M. Maabreh, B. Alsinglawi, M. Elkhodr, and N. Alsaedi, "Enhancing robots navigation in internet of things indoor systems," *Computers*, vol. 10, no. 11, p. 153, 2021.
- [61] O. Karasakal, L. Kandiller, and N. E. Özdemirel, "A branch and bound algorithm for sector allocation of a naval task group," *Naval Research Logistics (NRL)*, vol. 58, no. 7, pp. 655–669, 2011.
- [62] K. Solovey, S. Bandyopadhyay, F. Rossi, M. T. Wolf, and M. Pavone, "Fast near-optimal heterogeneous task allocation via flow decomposition," in *IEEE ICRA*, 2021, pp. 9117–9123.
- [63] I. P. Latini, W. E. Barioni, M. Teixeira, F. Neves-Jr, and L. V. R. de Aruda, "A mixed integer linear programming model for the distribution of transport tasks among heterogeneous robots in a factory," in *IEEE International Conference on Industrial Technology*, 2024, pp. 1–6.
- [64] J. G. Martin, J. R. D. Frejo, R. A. García, and E. F. Camacho, "Multi-robot task allocation problem with multiple nonlinear criteria using branch and bound and genetic algorithms," *Intelligent Service Robotics*, vol. 14, no. 5, pp. 707–727, 2021.
- [65] Y. Y. Liao and K. Ryu, "Genetic algorithm-based task allocation in multiple modes of human-robot collaboration systems with two cobots," *The International Journal of Advanced Manufacturing Technology*, vol. 119, no. 11, pp. 7291–7309, 2022.
- [66] E. Xidias and P. Zacharia, "Balanced task allocation and motion planning of a multi-robot system under fuzzy time windows," *Engineering Computations*, vol. 41, no. 5, pp. 1301–1326, 2024.
- [67] J. Wang, Y. Cai, Y. Zhou, R. Wang, and C. Li, "Discrete particle swarm optimization based on estimation of distribution for terminal assignment problems," *Computers & Industrial Engineering*, vol. 60, no. 4, pp. 566–575, 2011.
- [68] N. Wang, X. Yang, T. Wang, J. Xiao, M. Zhang, H. Wang, and H. Li, "Collaborative path planning and task allocation for multiple agricultural machines," *Computers and Electronics in Agriculture*, vol. 213, p. 108218, 2023.
- [69] J. C. Amorim, V. Alves, and E. P. de Freitas, "Assessing a swarm-GAP based solution for the task allocation problem in dynamic scenarios," *Expert Systems with Applications*, vol. 152, p. 113437, 2020.
- [70] J. Del Ser, M. Matinmikko, S. Gil-López, and M. Mustonen, "Centralized and distributed spectrum channel assignment in cognitive wireless networks: a harmony search approach," *Applied Soft Computing*, vol. 12, no. 2, pp. 921–930, 2012.
- [71] M. Umlauf, M. Gojkovic, K. Harshina, K. Majbour, and M. Schranz, "Bees, bats and glowworms: Swarm algorithms for optimizing industrial plants from the bottom-up," in *International Conference on Agents and Artificial Intelligence*. Springer, 2023, pp. 3–25.
- [72] H. Lin, S. Garg, J. Hu, G. Kaddoum, M. Peng, and M. S. Hossain, "Blockchain and deep RL empowered spatial crowdsourcing in software-defined internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3755–3764, 2020.
- [73] X. Zhao, Q. Zong, B. Tian, B. Zhang, and M. You, "Fast task allocation for heterogeneous unmanned aerial vehicles through RL," *Aerospace Science and Technology*, vol. 92, pp. 588–594, 2019.
- [74] W. Chang, W. Lizhen, Y. Chao, W. Zhichao, L. Han, and Y. Chao, "Coactive design of explainable agent-based task planning and deep reinforcement learning for human-UAVs teamwork," *Chinese Journal of Aeronautics*, vol. 33, no. 11, pp. 2930–2945, 2020.