

Review

Not peer-reviewed version

---

# Optimizing Coalition Formation Strategies for Scalable Multi-Robot Task Allocation: A Comprehensive Survey of Methods and Mechanisms

---

[Krishna Arjun](#)\*, [David Parlevliet](#), [Hai Wang](#), [Amirmehdi Yazdani](#)

Posted Date: 8 May 2025

doi: 10.20944/preprints202505.0606.v1

Keywords: Multi-robot; multi-robot task allocation; Coalition formation; reinforcement learning; convergence; evolutionary optimization



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Review

# Optimizing Coalition Formation Strategies for Scalable Multi-Robot Task Allocation: A Comprehensive Survey of Methods and Mechanisms

Krishna Arjun \*, David Parlevliet, Hai Wang and Amirmehdi Yazdani

School of Engineering and Energy, Murdoch University, WA, Australia

\* Correspondence: Krishna.arjun@murdoch.edu.au

**Abstract:** In practical applications, the utilization of multi-robot systems (MRS) is extensive and spans various domains such as search and rescue operations, mining operations, agricultural tasks, and warehouse management. The surge in demand for MRS has prompted extensive exploration of Multi Robot Task Allocation (MRTA). Researchers have devised a range of methodologies to tackle MRTA problems, aiming to achieve optimal solutions, yet there remains room for further enhancements in this field. Among the complex challenges in MRTA, the identification of an optimal coalition formation (CF) solution stands out as one of the NP-hard (Nondeterministic Polynomial time) problems. CF pertains to the effective coordination and grouping of agents or robots for efficient task execution achieved through optimal task allocation. In this context, this paper delivers a succinct overview of dynamic task allocation and CF strategies. It conducts a comprehensive examination of diverse strategies employed for MRTA. The analysis encompasses the advantages, disadvantages, and comparative assessments of these strategies with a focus on CF. Furthermore, this study introduces a novel classification system for prominent task allocation methods and compares these methods with simulation analysis. The fidelity and effectiveness of the proposed CF approach are substantiated through comparative assessments and simulation studies.

**Keywords:** multi-robot; multi-robot task allocation; coalition formation; reinforcement learning; convergence; evolutionary optimization

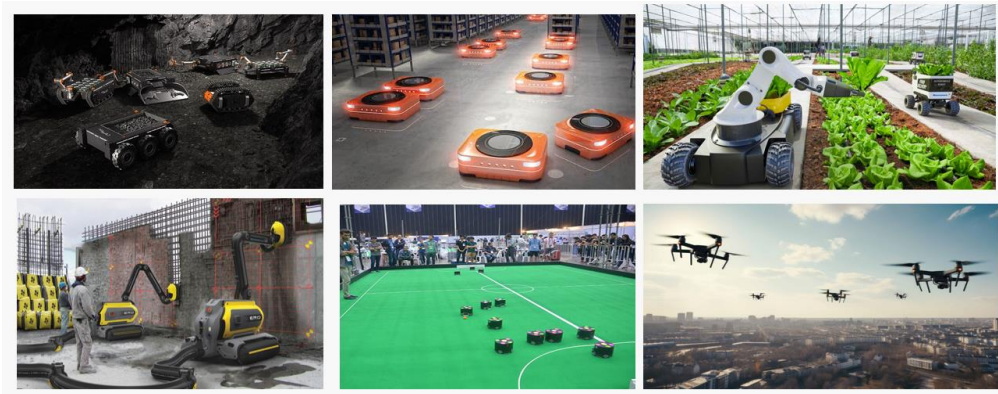
## 1. Introduction

In the era marked by the emergence of Industry 4.0, there has been a remarkable increase in the demand for autonomous systems and Artificial Intelligence. The field of robotics has experienced widespread popularity across various sectors of society. As a result, the concept of multi-robot systems (MRS) has gained significant attention, surpassing that of individual robots. A multi-robot system comprises a group of robots working together to achieve a predetermined objective. A related term, like MRS, is the multi-agent system (MAS), which consists of a collective of agents collaborating to accomplish a specific goal [1]. The key distinction between MRS and MAS lies in the former involving physical robots as agents, whereas the latter incorporates agents represented as software entities. MRS offer numerous advantages over individual robots:

- (a) MRS enables parallel task execution, leading to accelerated goal attainment.
- (b) Heterogeneity in robot capabilities can be accommodated within MRS.
- (c) MRS effectively handles tasks distributed across large spatial domains.
- (d) Inherent robustness in fault tolerance is a characteristic feature of MRS.

The taxonomy of MRS divides into two main categories: homogeneous assemblies, consisting of robots with similar capabilities, and heterogeneous assemblies, comprising robots with varying proficiencies. Additionally, this classification extends to cooperative ensembles, where robots collaborate to achieve goals, and competitive configurations, where robots compete for dominance. In recent times, there has been a significant increase in the demand for MRS in practical applications

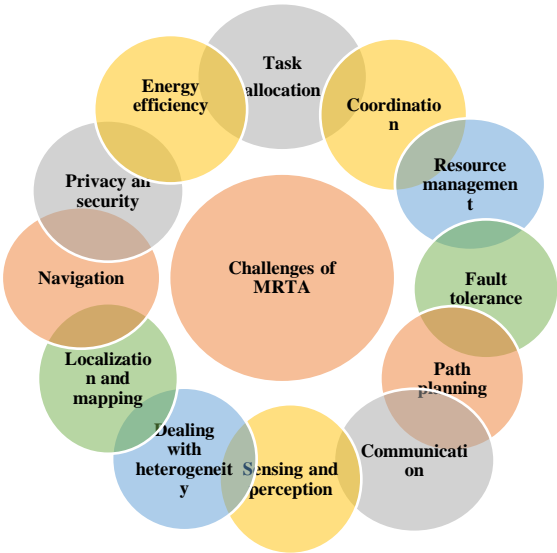
[2]. This growing demand has led to the integration of MRS into various domains, including mining, warehouse logistics, agricultural field operations, and even recreational and entertainment sectors. However, this widespread adoption of MRS spans across numerous sectors globally [3]. As this trend continues, addressing the existing challenges inherent to MRS becomes increasingly critical.



**Figure 1.** Real world applications of MRS [101–106].

MRS pose a range of intricate challenges across technological, logistical, and operational spheres, as depicted in Figure 2. Key challenges include coordinating robot actions to avoid collisions or redundancy, ensuring reliable communication, and efficient task allocation [4]. Moreover, addressing faults, maintaining accurate environmental perception, managing shared resources while upholding privacy, security, and ethical human-robot interactions are significant hurdles. Energy efficiency is crucial, especially for battery-powered robots. These challenges collectively render the development and operation of multi-robot systems multifaceted.

Task allocation stands out as particularly crucial among these challenges due to its pivotal role in optimizing resource utilization, enhancing system efficiency, and facilitating effective collaboration among robots. Efficient task allocation in multi-robot systems offers various benefits. Firstly, it optimizes resource usage, leading to cost savings and improved system performance. Secondly, by ensuring tasks are executed effectively and promptly, task allocation enhances overall system efficiency. Thirdly, it facilitates scalability, enabling seamless integration of additional robots or tasks without compromising efficiency [5]. Moreover, task allocation contributes to system robustness by allowing adaptation to dynamic environments and changing task demands. Lastly, it promotes collaboration among robots, reducing conflicts and redundancy while enhancing system coordination.

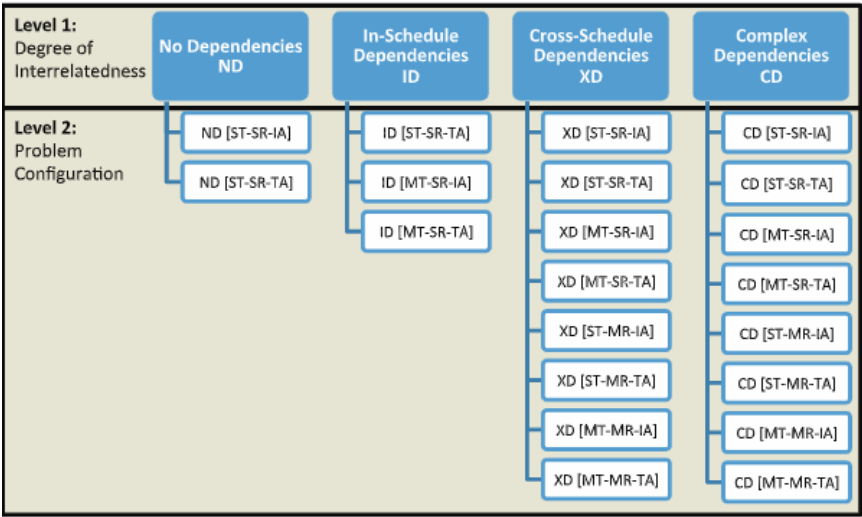


**Figure 2.** Challenges of MRS.

1.1. Multi-Robot Task Allocation (MRTA)

Task allocation within a multi-robot system involves determining which robots should perform specific tasks to achieve overarching system objectives, aiming for coordinated team behavior. While some systems, like certain biologically inspired robotic setups, exhibit coordinated team behavior through local interactions among team members and the environment, known as implicit or emergent coordination [6], some other lies on explicit or intentional cooperation. In this approach, tasks are directly assigned to robots or sub-teams, constituting the problem of multi-robot task allocation.

The study reported in [7] proposed a categorization framework for MRTA problems based on three axes. The first axis distinguishes between single-task (ST) robots, capable of executing only one task at a time, and multi-task (MT) robots, capable of handling multiple tasks simultaneously. The second axis categorizes tasks as Single-Robot (SR) tasks, requiring one robot for completion, or Multi-Robot (MR) tasks, necessitating the involvement of multiple robots. The third axis differentiates between Instantaneous Assignment (IA) problems, which involve immediate task allocation without consideration for future assignments, and Time-Extended Assignment (TA) problems, where robots are allocated, tasks based on a predetermined schedule encompassing both current and future allocations [7].The study delineates the MRTA quandary into eight distinct typologies known as two level task allocation taxonomy (iTax) classification, as illustrated in Figure 3 [8]. This scheme rests on the foundation of interdependent resources and constraints. Employing the iTax framework, the MRTA conundrum was segmented into four distinct categories: those devoid of dependencies, those governed by scheduled dependencies, those subjected to cross-scheduled dependencies, and finally, those characterized by complex dependencies [9].



**Figure 3.** iTax: a two-level task allocation taxonomy [12].

The limitations of the iTax have driven the exploration of alternative strategies like auction-based and optimization-based approaches to tackle MRTA problems. iTax's complexity arises from its division of task allocation into high-level and low-level processes, potentially complicating system design and coordination. Scalability issues may arise with iTax as the number of robots and tasks increases, leading to inefficiencies in allocation and coordination. Additionally, iTax may yield sub-optimal solutions due to the separation of high-level decisions from low-level motion planning considerations. Consequently, auction-based approaches, inspired by auction dynamics, and optimization-based approaches, relying on mathematical formulations, have emerged as alternatives. Auction-based approaches enable dynamic task allocation through robots bidding based on their



capabilities, while optimization-based approaches offer systematic solutions considering resource utilization and task efficiency. These alternatives aim to address iTax's limitations and enhance MRTA problem-solving effectiveness.

Considering factors such as solution optimality, allocation timing, and problem-specific constraints, two crucial task allocation strategies have been developed: auction-based approaches [10] and optimization-based approaches [11]. Auction-based approaches draw inspiration from the dynamics of auction-bid systems observed in societal contexts. In this approach, robots submit bids reflecting their current state values to compete for tasks. The task is then assigned to the robot offering the most suitable bid, representing a merit-based selection process. Conversely, optimization approaches involve devising mathematical solutions to tackle task allocation challenges. Additionally, behavior-based methodologies [12] have garnered considerable scholarly attention for their ability to adapt to the dynamic behavior of systems. More recently, innovative dynamic task allocation strategies have emerged to address complex constraints associated with various uncertain conditions. The growing significance of Artificial Intelligence (AI) and learning-based approaches has become prominent in this domain [13]. Researchers are increasingly turning their attention to these methods, motivated by their potential to provide comprehensive autonomous and optimal solutions to MRTA challenges. This trend reflects the evolving landscape of the field.

MRTA optimizes system performance by efficiently distributing tasks among multiple robots, crucial for various applications such as resolving complex challenges like the Multiple Traveling Salesman Problems (mTSP), Vehicle Routing Problems (VRP), Job Scheduling Problems (JSP), Team Orienting Problems (TOP), and Dial-a-Ride Problems (DARP). In mTSP, multiple robots solve a variant of the Traveling Salesman Problem, optimizing routes to minimize travel distance and costs [14]. VRP involves efficiently routing tasks among robots, resembling delivery vehicles, considering constraints like capacities and time windows [15]. JSP focuses on scheduling tasks among robots to minimize make span, critical for industrial automation and warehouse management [16]. TOP dynamically forms and optimizes teams of robots to accomplish tasks collaboratively, crucial for search and rescue missions [17]. DARP coordinates multiple robots to provide transportation services, relevant for ride-sharing and public transportation systems, emphasizing the optimization of passenger routing and vehicle allocation [18].

The coalition formation (CF) remains a crucial but unaddressed aspect in a wide array of multi-robot task allocation problems, such as mTSP, VRP, Job JSP, TOP, and DARP. These problems involve multiple robots or agents tasked with optimizing various objectives. For instance, in mTSP, the challenge is to have multiple salesmen (robots) efficiently visit a set of locations while minimizing the total route length. The establishment of coalitions can facilitate the equitable distribution of tasks, leading to a reduction in the overall travel distance. In VRP, the objective is to deliver goods to multiple customers while minimizing transportation costs. Here, coalitions play a vital role in route and resource sharing to achieve cost-effective solutions. Similarly, in JSP, multiple robots must execute a variety of tasks while optimizing time and resource utilization. The formation of coalitions aids in workload balancing and reduces the make-span. In TOP, the focus is on organizing robots into teams to accomplish specific tasks effectively. Coalitions are instrumental in the formation of teams that maximize task performance. Lastly, DARP involves multiple robots providing transportation services to passengers with pickup and drop-off requests. In this context, coalitions help in passenger sharing and route optimization.

In conclusion, CF is an indispensable aspect of addressing the challenges posed by MRTA across various domains. This paper reviews MRTA algorithms, assesses their support for CF, and seeks to determine the most effective methods for achieving optimal CF.

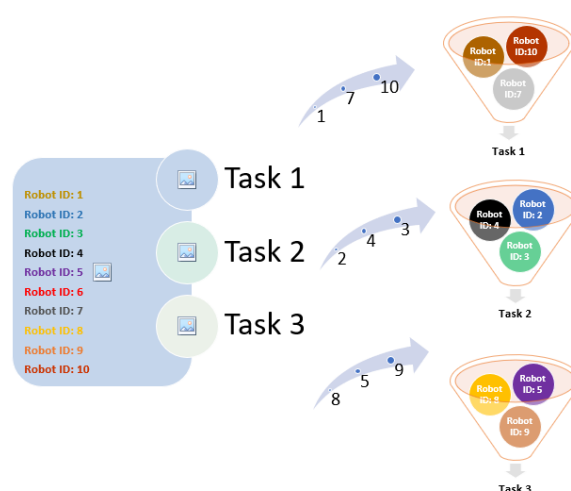
### 1.2. Coalition Formation (CF)

In the realm of robotics or multi-agent systems, the assembly of separate groups of robots or agents, each assigned specific tasks, constitutes coalition formation presented in Figure 4 [19]. Task allocation, within this framework, arises as a fundamental aspect of CF, involving the assignment of

tasks to robots based on their capabilities. Optimal CF requires maximizing overall team performance while minimizing task completion time and resource consumption [20].

Various algorithms aimed at facilitating efficient collaboration among robots through CF have been devised. Examples include the Contract Net Protocol (CNP) [21], Consensus-Based Bundle Algorithm (CBBA) [22], Task Allocation Via Iterative Bargaining (TAIB) [23], Stable Marriage Algorithm (SMA) [24], Dynamic Distributed Coalition Formation (DDCF), as well as the Merge and Split Approaches, Combinatorial Auctions, Core Selecting Combinatorial Auction (CCA), and the Genetic Algorithm (GA) for CF [25]. These algorithms incorporate negotiation mechanisms, enabling robots to form coalitions based on their individual competencies and preferences.

In the context of the CNP, tasks are disseminated to participating robots by a central coordinator. Interested robots then submit bids for tasks, and the coordinator selects winning bids to form coalitions. CBBA involves robots engaging in communication and negotiations through a consensus-based framework, facilitating deliberation on coalition assignments and task allocation. In TAIB, robots negotiate with other entities to assess potential coalitions, iteratively refining preferences during CF. The SMA entails robots (male) proposing to potential coalitions (females), with coalitions accepting or rejecting proposals based on preferences. DDCF utilizes a distributed negotiation approach, enabling robots to adapt to changes in team composition. The Merge-and-Split algorithm allows existing coalitions to merge or split based on evolving task requirements and robot capabilities. Combinatorial auctions offer bundles of tasks or resources for bidding, with the auctioneer assigning winning combinations to maximize efficiency. Core CCA aims to determine stable and efficient task allocations among coalitions. Lastly, in GA, robot capabilities and preferences are represented as genes, subject to selection, crossover, and mutation operations to converge towards an optimal coalition structure [26].



**Figure 4.** Concept of coalition formation.

The domain of multi-robot CF presents various challenges, including managing uncertainty, partial observability [27], non-stationarity [28], communication disruptions, the heterogeneity of robot capabilities, and task interdependencies. Developing efficient coalition algorithms is a significant task, often requiring substantial computational resources and specialized expertise. Maintaining coordination within coalition-based interactions in dynamic environments can be complex, with synchronization of actions emerging as a crucial concern. Additionally, multiple coalitions may converge to undertake similar or overlapping tasks, potentially leading to inefficiencies [29,30]. Nevertheless, CF enables robots to share resources such as energy, computational power, and communication bandwidth, enhancing resource utilization [31]. This sharing facilitates flexibility in team configuration adaptation in response to changing environmental

conditions or task requirements. Such adaptive capability enhances system resilience against unforeseen challenges and improves overall robustness.

This article is a revised and expanded version of a paper entitled “Analyzing multi-robot task allocation and coalition formation methods: A comprehensive study,” which was presented at the 1st International Conference on Advanced Robotics, Control, and Artificial Intelligence (ARCAI), Australia, December 2024. The rest of this paper offers a concise taxonomy of various strategies utilized in MRTA and CF. In Section 2, detailed insights into each strategy are provided along with a comparative analysis of different approaches within each strategy and a higher-level comparison of strategies is conducted based on relevant factors. Section 3 focuses on algorithm formulation and presents simulation results, while Section 4 serves as the discussion segment, summarizing the comprehensive review and highlighting the key findings. Section 5 gives our conclusion.

2. MRTA Classification

Figure 5 depicts an arrangement presenting a categorization of strategies pertaining to MRTA. Within this taxonomy, four overarching classifications are discernible: behavior-based, market-based, optimization-based, and learning-based. Each of these classifications introduces a collection of prominent and well-established methodologies to address MRTA challenges, representative of standard paradigms. While certain researchers are modifying or extending these established approaches, others are integrating aspects of these methodologies to engender hybrid strategies. The four main categories of MRTA strategies are Behavior-based MRTA, Market-based MRTA, Optimization-based MRTA and Learning-based MRTA.

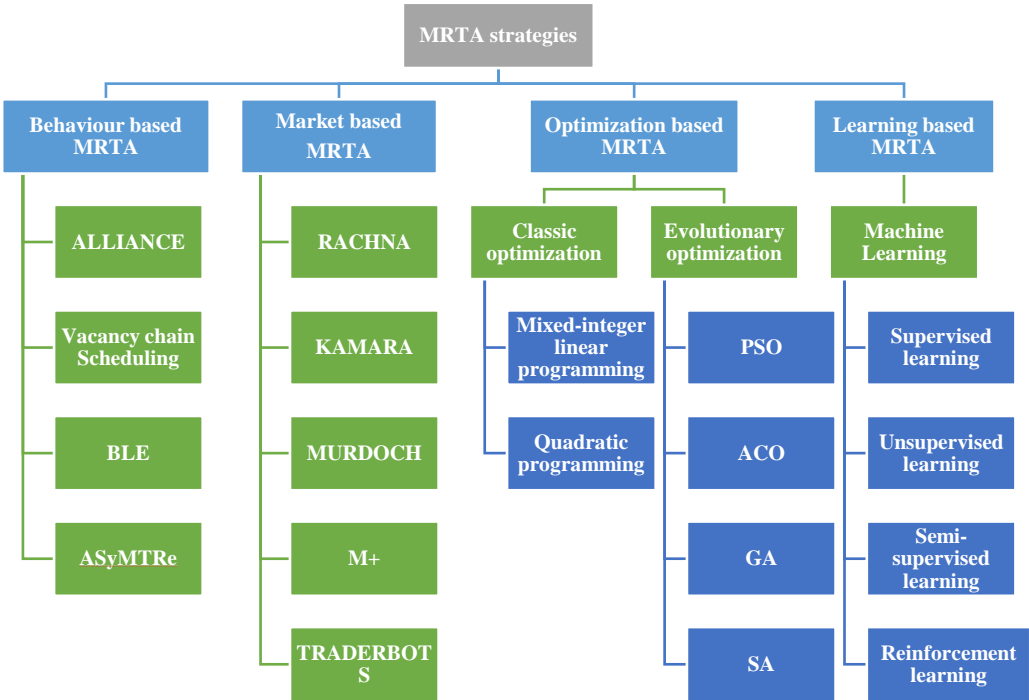


Figure 5. Classification chart for MRTA strategies.

2.1. Behavior-Based MRTA

Behavior-based MRTA represents a distinctive strategy characterized by its entirely reactive response to dynamic problem scenarios, as suggested by its nomenclature. This approach is structured upon a dual-tier architecture, comprising a foundational lower level and a supervisory

stratum [32]. The lower level includes tasks such as navigation, obstacle avoidance, and task-switching, while the supervisory level handles task identification and inter-robot communication within the team. To enhance resilience, a problem-specific layer augments the lower-level behavior [33]. Figure 5 illustrates that behavior-based MRTA encompasses four methodological approaches: ALLIANCE, vacancy chain scheduling, Broadcast of Local Eligibility (BLE), and Automated Synthesis of Multi-Robot Task Solutions through Software Reconfiguration (ASyMTRe) [6].

#### 2.1.1. Alliance

The Alliance architecture, known for its behavior-based and fully distributed principles, enables MRS to adapt to diverse circumstances by organizing behaviors tailored to high-level tasks [34–36]. Key features include robot impatience and acquiescence within coalitions, which impact collaboration by prioritizing individual goals or yielding excessively, respectively. Balancing these behaviors is crucial for effective collaboration [37–39]. Alliance supports task allocation, resource sharing, and decision-making among dynamically assembled coalitions, enhancing communication and information exchange for optimized performance [40,41]. Attributes include distributed decision-making, reactive behavior, emergent coalition behavior, robustness, scalability, and adaptability.

#### 2.1.2. Vacancy Chain Scheduling

The Vacancy Chain System (VCS) models scheduled by creating a cascade effect through job promotions, akin to bureaucratic structures. This method addresses scheduling intricacies and group dynamics using microscopic and macroscopic approaches [42]. While effective in spatial domains, VCS's performance varies with individual robot capabilities. Its strength lies in its versatility across problems, requiring minimal problem-specific information. Key attributes include distributed decision-making, reactive behavior, emergent system behavior, robustness against robot unavailability, scalability, and adaptability through scheduling algorithm adjustments [43,44].

#### 2.1.3. Broadcast of Local Eligibility (BLE)

The Broadcast of Local Eligibility (BLE) mechanism compares a task's local eligibility with the highest eligibility among similar behaviors in other robots, following the Port Arbitrated Behavior (PAB) paradigm [45]. BLE operates by suppressing peer behaviors when a robot's local eligibility surpasses others, signaling its claim to tasks. In the absence of inhibitory interactions, other robots assume task responsibility. BLE's scalability depends on communication bandwidth, with attributes including distributed decision-making, reactive behavior, emergent behavior, robustness against failures, scalability, adaptability, and local communication for sharing eligibility information among neighbors [46,47].

#### 2.1.4. Automated Synthesis of Multi-Robot Task Solutions through Software Reconfiguration (ASyMTRe)

ASyMTRe automatically reconfigures schema connections across and within robots by linking environmental, perceptual, and motor control schemas, facilitating effective multi-robot behaviors aligned with team objectives [48]. It addresses task-related challenges in diverse robot teams, enabling the synthesis of new task solutions and sharing sensory information among networked robots to assist less capable ones [49]. ASyMTRe exhibits distributed decision-making, reactive behavior, emergent behavior, robustness, scalability, and adaptability, allowing robots to autonomously make decisions, adjust to environmental changes, and withstand failures [50].

As previously mentioned, ALLIANCE, Vacancy chain, BLE, and ASyMTRE are four key algorithms falling under the umbrella of behavior-based MRTA. Despite their shared behavior-based nature, they exhibit distinct attributes and advantages and disadvantages. Tables 1 and 2, below present a comprehensive comparative analysis of these algorithms based on their characteristics. The



comparison of these attributes underscores ALLIANCE's proficiency in effectively managing CF in Multi-Robot Systems (MRS).

**Table 1.** Efficiency and trade-offs of various approaches in behavior-based MRTA.

Algorithm	Efficiency	Advantages	Disadvantages
Alliance	High	Scalable, adaptable to dynamic environments, provide a higher degree of stability in coalition.	Requires effective communication and coordination
Vacancy Chain	Medium	Low communication overhead, stable coalitions	Limited scalability, sensitive to changes in the team
Broadcast Local Eligibility	Medium to high	Efficient, distributed, Low communication	Tends to form smaller coalitions
ASyMTRE	High	Adaptive, efficient task allocation	Requires sophisticated negotiation mechanisms

**Table 2.** Comparison of different algorithms under behavior-based MRTA.

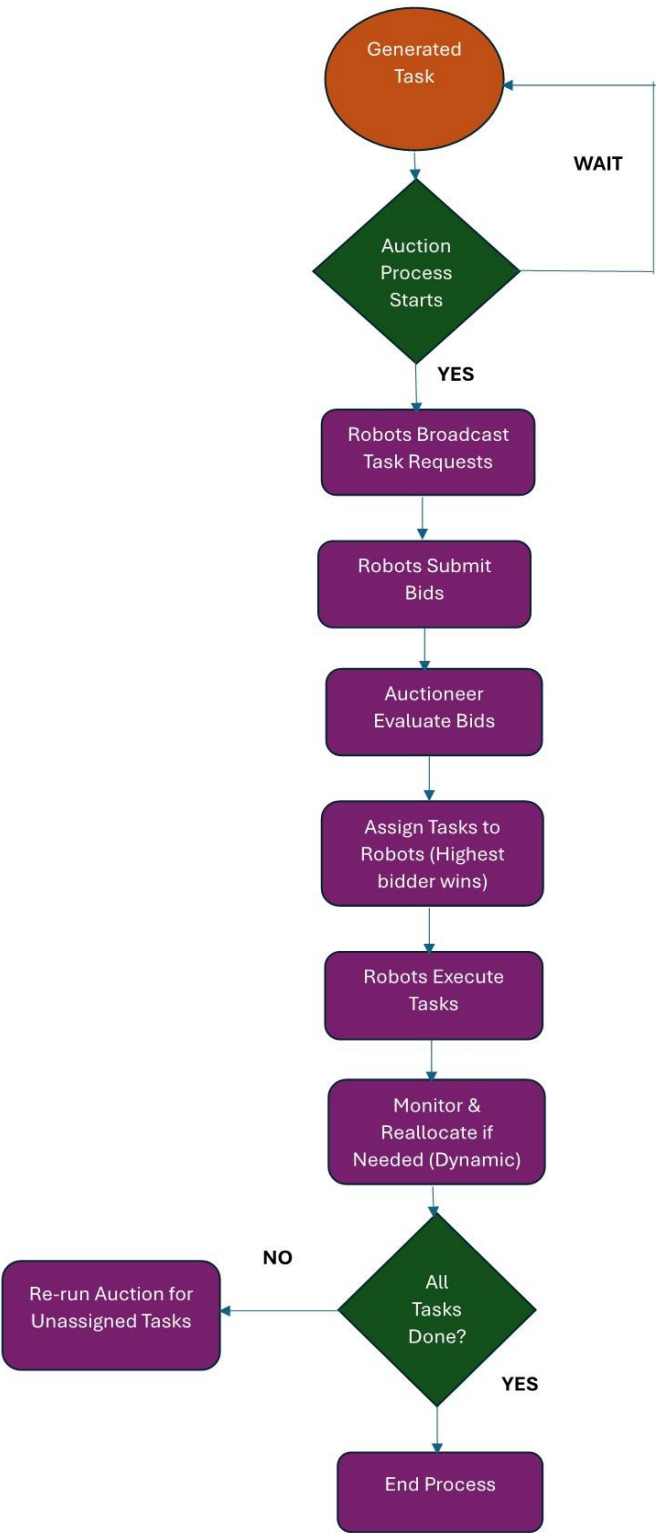
Characteristics	Alliance	Vacancy chain	BLE	ASyMTRE
Homogenous/Heterogenous	Heterogeneous	Homogenous robots	Heterogeneous	Heterogeneous
Optimal allocation	Guarantee optimal allocation	Guarantee (Minimal)	Does not Guarantee	Guarantee (Minimal)
Cooperation	Strongly cooperative	Weak cooperation	Strongly cooperative	Strongly cooperative
Communication	Strong	Limited	Strong	Limited
Hierarchy	Fully distributed	Not fully distributed	Fully distributed	Not fully distributed

Task reassignment	Possible through coalition reconfiguration)	Possible (via vacancy announce ment)	(Possible based on dynamic eligibility)	(Possibly based on genetic optimization)
----------------------	---	--	---	---

2.2. Market-Based MRTA

The foundational concept of market based MRTA draws inspiration from the principles of auctions and bidding. Various market-based methodologies are crafted by adapting the core auction-bidding procedure with the incorporation of incentives and penalties. Figure 6 illustrates the fundamental genesis of market based MRTA.

Market-oriented strategies focus on utility functions, quantifying agents' capacity to evaluate their interest in particular tasks for potential exchange. In MRTA systems, these functions clarify the alignment between a robot's skills and task requirements [19]. Market-based approaches in multi-robot task allocation exhibit distinctive attributes that enhance their efficacy. By centering on utility functions, agents can gauge preferences for tasks, facilitating efficient task trading. Auction mechanisms introduce competitive bidding, motivating agents to bid based on perceived task values [51]. Incentive-driven behavior, through rewards and penalties, ensures optimal task allocation and fosters cooperation. Decentralized decision-making empowers agents to autonomously engage in task assignments. The adaptability of market-based strategies allows responses to changing conditions by adjusting utility functions and allocation mechanisms. These methods are flexible, accommodating various tasks and heterogeneous robot capabilities [52]. Through bid submissions and communication, agents share information, enhancing decision-making. The scalability of market-based systems effectively manages large-scale multi-robot scenarios, resulting in robust allocations that maximize overall utility, contributing to resource-efficient task distribution. Collectively, these attributes equip market-based approaches to effectively address multi-robot task allocation challenges.



**Figure 6.** A basic model of market-based MRTA.

2.2.1. RACHNA

RACHNA, an ascending auction-based task allocation protocol that allows task preemption, introduces market-based approaches to CF in competitive environments but faces challenges like unnecessary task reassignments, impacting performance due to switching overhead [53]. Designed for CF, RACHNA addresses heuristic-driven methods' limitations by allowing tasks to compete for robot resources and autonomously adjusting sensor values based on supply and demand, without

constraints on coalition size [54,55]. It frames CF as a multi-unit combinatorial auction, enabling efficient management of CF through bids on combinations of goods [56].

2.2.2. KAMARA (KAMRO’s Multi-Agent Robot Architecture)

KAMARA combines centralized and distributed frameworks for the Karlsruhe Autonomous Mobile Robot (KAMRO). This architecture enhances control and integration across diverse aspects of distributed intelligent robot systems and their individual components, accommodating various forms of cooperation among interconnected agents, such as closed kinematic chains or camera-manipulator couplings [57,58].

2.2.3. MURDOCH

MURDOCH involves an Auctioneer and Bidder in a swift task allocation method through initial-price, one-round auctions, assigning tasks to the highest bidder based on computed bid values reflecting agents' capabilities [59]. It introduces a robust MRTA algorithm adept at handling robot malfunctions and communication breakdowns, allowing task reassignment and facilitating collaboration among heterogeneous robots [60]. The Auctioneer initiates auctions upon task arrival, selects winning bidders, and monitors task execution, while bidders compute utility metrics, submit bids, and execute tasks upon winning, using a publish-subscribe communication model for coordination [61–64].

2.2.4. M+

M+ is a distributed multi-robot cooperation scheme integrating mission planning, task refinement, and cooperative mechanisms from the Contract Net Protocol framework within the M+ unified system, designed for integration using the LAAS Architecture [65,66]. It includes M+ task allocation, cooperative response to contingencies, and task execution operations, refining and assigning tasks through negotiation mechanisms. The cooperative reaction activity handles task execution failures by updating the world state, facilitating information exchange, overseeing (re)planning, and coordinating assistance, while the execution function manages task control and synchronization among robots [67,68].

2.2.5. TraderBots

The TraderBots methodology leverages market economies' benefits for effective multi-robot coordination in dynamic scenarios through decentralized decision-making and task allocation within coalitions, demonstrated in simulations and physical implementations [69–71]. It excels in CF by employing a market-based framework, facilitating efficient task allocation based on robots' capabilities and task compatibility, adapting to dynamic environments, and fostering information exchange among robots for informed decision-making. TraderBots is robust against failures and communication disruptions, showcasing versatility and scalability for diverse multi-robot scenarios, proving its practicality for effective CF [72].

All these methodologies fall within the realm of market-based task allocation and are founded on the common principle of employing market mechanisms to distribute tasks among robots. However, they diverge in terms of bid generation methods and the precise algorithms employed to optimize the task allocation process. The Table 3 presented below illustrates the comparison of five market-based approaches according to their distinctive attributes.

**Table 3.** Comparison of different algorithms under market based MRTA.

Characteristics	RACHNA	KAMARA	MURDOCH	M+	TraderBots
-----------------	--------	--------	---------	----	------------

<b>Market-based</b>	Negotiation based	Market-based	Market-based	Negotiation based	Auction based
<b>Bidding method</b>	Uses a genetic algorithm to optimize bids	Bids are based on utility functions that consider the cost and quality of the task	Bids based on a simple cost function	Form coalitions to bid on tasks together	Bids are based on a reinforcement learning algorithm
<b>Homogenous/Heterogeneous</b>	Heterogeneous	Heterogeneous	Heterogeneous	Heterogeneous	Homogeneous robots
<b>Fault tolerance</b>	Not fault-tolerant	Fault-tolerant	Not fault-tolerant	Fault-tolerant	Fault-tolerant
<b>Optimal allocation</b>	Can guarantee depending on the fitness function	Can guarantee based on the utility function	Not Guaranteed	Can guarantee based on the coalition formation algorithm	Can guarantee
<b>Cooperation</b>	Cooperative	Cooperative	Strong cooperation	Cooperative	Strongly cooperative
<b>Communication</b>	Limited (Global communication)	Limited (Local communication)	Strong (Global communication)	Strong (Local communication)	Strong (Local communication)
<b>Hierarchy</b>	Distributed	Hybrid	Distributed (Loosely coupled)	Fully distributed	Combination of a distributed and centralized approach
<b>Task reassignment</b>	Not Possible	Possible	Not possible	Possible	Possible
<b>Complexity</b>	Moderate	Moderate	Simple	High	High
<b>Cost</b>	Moderate	Moderate	Low	High	High
<b>Scalability</b>	Limited	Highly scalable	Limited	Highly scalable	Highly scalable
<b>Coalition formation</b>	Yes	Possible	Yes, and dynamically adaptable	Yes, and dynamically adaptable	Yes



2.3. Optimization-Based MRTA

Optimization-driven methods strive to address the MRTA challenge by framing it as an optimization dilemma, seeking the best task-to-robot assignment. These methods primarily fall into two categories: conventional optimization methods and evolutionary optimization strategies.

Traditional optimization methods encompass mathematical frameworks, among which are Mixed Integer Linear Programming (MILP) and Quadratic Programming (QP), frequently employed. In contrast, evolutionary optimization approaches such as Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), GA, and Simulated Annealing (SA) have gained popularity [73]. While MILP and QP are favored for their potential to identify globally optimal solutions, their computational complexity can escalate with larger problem scales [74]. On the other hand, PSO, ACO, GA, and SA serve as metaheuristic optimization methods, capable of managing multiple objectives and non-linear constraints, although they might converge to local optima.

2.3.1. Traditional Optimization

MILP and QP are prevalent traditional methods for solving multi-robot task allocation challenges. MILP involves linear objective functions and constraints, while QP deals with non-linear formulations. A significant challenge with MILP is its scalability due to increased complexity and variable quantities [75]. In contrast, QP controllers are becoming benchmarks for managing complex objectives in robots with multiple joints, such as humanoids, proving effective in position control, torque regulation, multi-robot coordination, and force management [76].

2.3.2. Evolutionary Optimization

PSO operates as a population-centric search algorithm inspired by social behavior, iteratively exploring optimization problems within a designated search domain, making it suitable for robots with limited capabilities [77–80].

ACO, initiated by Dr. Marco Dorigo, emulates ant foraging behavior and finds applications in routing and scheduling, functioning as a probabilistic strategy for identifying favorable pathways across graphs within swarm intelligence methodologies [81–85].

GA explores optimization challenges through competitive selection, recombination, and mutation within a population of solutions, generating novel solutions biased towards favorable regions of the solution space [86,87].

SA mimics physical annealing transformations in solids, gradually reducing temperature to converge towards the most favorable state, using temperature-dependent random factors to escape local minima and reach lower energy configurations [88–90].

Optimization-based MRTA approaches: Methods like PSO, ACO, GA, SA, LP, and QP exhibit diverse attributes shaping their problem-solving dynamics.

- PSO seeks global optima while navigating exploration and exploitation trade-offs.
- ACO emphasizes pheromone-guided exploration and solution construction.
- GA balances diversity through mutation and convergence via crossover.
- SA transitions from high-temperature exploration to low-temperature exploitation.
- LP targets linear relationships, and QP handles quadratic ones, both optimized for resource allocation.

Each approach's distinct characteristics equip them to excel in various problem domains, addressing the intricacies of task allocation scenarios. A comparison of these approaches, considering various attributes, is presented in tables 4 and 5 below.

**Table 4.** Comparative overview of optimisation-based approaches.

Approach	Optimization technique	Advantages	Disadvantages
----------	------------------------	------------	---------------

PSO [75–78]	Swarm intelligence	<ul style="list-style-type: none"> <li>• Easy implementation</li> <li>• Efficient in solving continuous problems.</li> <li>• Can handle multiple objectives.</li> <li>• Can handle non-linear constraints.</li> <li>• Fast convergence</li> </ul>	<ul style="list-style-type: none"> <li>• Premature convergence.</li> <li>• Weak local search ability.</li> <li>• Require extensive parameter tuning.</li> </ul>
ACO [79–83]	Swarm intelligence	<ul style="list-style-type: none"> <li>• Rapid discovery of reasonable solutions.</li> <li>• Efficient in solving TSP problems and other discrete problems.</li> <li>• Can handle non-linear constraints.</li> <li>• Can search global optima.</li> <li>• Can handle the dynamic environment.</li> </ul>	<ul style="list-style-type: none"> <li>• Premature convergence.</li> <li>• Weak local search ability.</li> <li>• Probability distribution changes Iterations.</li> <li>• Not effective in solving continuous problems.</li> <li>• High computational cost</li> </ul>
GA [84,85]	Evolutionary	<ul style="list-style-type: none"> <li>• Exchange information (Crossover or mutation)</li> <li>• Efficient in solving continuous problems.</li> <li>• Can search global optima.</li> <li>• Can handle multiple objectives</li> </ul>	<ul style="list-style-type: none"> <li>• Premature convergence.</li> <li>• Weak local search ability.</li> <li>• High computational effort.</li> <li>• Difficult to encode a problem.</li> </ul>
SA [86–88]	Stochastic	<ul style="list-style-type: none"> <li>• Rapid discovery of solutions.</li> <li>• Applicable to large data sets.</li> <li>• Can handle a dynamic environment.</li> <li>• Can escape local optima.</li> <li>• Can handle dynamic environments</li> </ul>	<ul style="list-style-type: none"> <li>• A large number of iterations may be required to achieve an optimal solution.</li> <li>• It does not guarantee that an optimal solution will be found.</li> <li>• Slow convergence</li> <li>• High computational cost</li> </ul>
MILP [72,73]	Mathematical Programming	<ul style="list-style-type: none"> <li>• Efficient in solving linear and non-linear constraints.</li> <li>• You can find global optima.</li> <li>• Can handle multiple objectives</li> </ul>	<ul style="list-style-type: none"> <li>• High computational effort.</li> <li>• May not handle dynamic environments.</li> <li>• May not scale well to significant problems</li> </ul>

QP [74]	Mathematical Programming	<ul style="list-style-type: none"><li>• Efficient in solving non-linear linear constraints</li></ul>	<ul style="list-style-type: none"><li>• High computational effort.</li><li>• Have restrictions in some problem situations.</li></ul>
---------	--------------------------	--	--

**Table 5.** Comparison of different algorithms under optimization-based MRTA.

Characteristics	Particle Swarm Optimization (PSO)/Ant Colony Optimization (ACO)/Genetic Algorithm (GA)/Simulated Annealing (SA)	Mixed Integer Linear Programming (M ILP)	Quadratic Programming (QP)
Fault tolerance	Robust to individual robot failures but not to system-wide failures	Not inherently fault tolerant	Not inherently fault tolerant
Optimal allocation	May converge to local optima and be able to handle multiple objectives	Can find globally optimal solutions, but computational complexity may increase with problem size.	Can find globally optimal solutions, but computational complexity may increase with problem size.
Scalability	Can handle significant problems efficiently but require extensive parameter tuning.	Small-medium sized problems	Can handle significant problems efficiently but require extensive parameter tuning
Task reassignment	Can handle by updating the objective functions and constraints	Can handle by updating the objective functions and constraints	Can handle by updating the objective functions and constraints
Coalition formation	Can handle by adding appropriate terms to the objective functions and constraints	Can handle by adding appropriate terms to the objective functions and constraints	Can handle by adding appropriate terms to the objective functions and constraints
Complexity	Can handle complex optimization problems with non-linearities and multiple objectives	Can handle linear and non-linear constraints.	Can handle linear and non-linear constraints.
Cost	It can be less expensive than MILP and QP but require extensive parameter tuning.	It can be expensive due to the computational complexity	It can be expensive due to the computational complexity

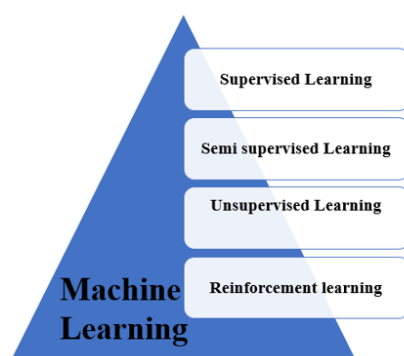
2.4. Learning-Based MRTA

Learning is fundamental to constructing solutions, representing a progression where an AI program enhances its understanding by observing its environment. Technically, AI learning mechanisms involve processing input-output pairs to deduce patterns for a designated function, enabling the anticipation of outputs for novel inputs. This paradigm allows robots to dynamically improve their task allocation strategies through learning from data and experience. By leveraging machine learning techniques, these systems adapt and optimize decisions over time, effectively navigating complex and changing environments [91]. This approach enhances resource utilization,

coordination, and overall efficiency in multi-robot systems by allowing them to autonomously refine task allocation behaviors based on real-world interactions and performance feedback.

#### 2.4.1. Machine Learning

Machine Learning, a subset of AI, enables machines to replicate intelligent human actions. It includes four main types: supervised, semi-supervised, unsupervised, and reinforcement learning [92]. These techniques allow computers to learn from data, facilitating informed decision-making and predictions. In multi-robot task allocation, machine learning enhances decision-making and adaptability, enabling dynamic optimization of task allocation strategies based on data-driven insights and historical performance [93]. By learning from interactions between robots, their environment, and tasks, multi-robot systems improve resource utilization, coordination, and system efficiency, enhancing scalability, flexibility, and autonomy in addressing task allocation challenges.



**Figure 7.** Structure of machine learning.

Supervised Learning trains algorithms on labeled data to predict specific outputs, iteratively refining models to discern patterns connecting input and output labels. In multi-robot task allocation, supervised learning maps behaviors to optimal allocations, using historical data to predict suitable assignments for new scenarios, optimizing resource utilization and system efficiency [94,95].

Semi-supervised Learning combines supervised and unsupervised learning, leveraging both labeled and unlabeled data to enhance model performance and generalization [96]. In multi-robot task allocation, it blends techniques to optimize resource utilization even with limited labeled data [97].

Unsupervised Learning operates without explicit guidance, uncovering patterns in data without predefined labels, grouping similar data points together to encapsulate datasets more compactly. In multi-robot task allocation, unsupervised learning distills insights from unstructured datasets, improving resource allocation strategies [98].

Reinforcement Learning involves models making sequential decisions to achieve objectives within unpredictable settings, continuously interacting with the environment and receiving rewards or penalties to adapt and refine behaviors over time. It is promising for optimizing resource utilization and improving collaboration in multi-robot task allocation systems [99–102].

Machine learning encompasses methodologies including supervised, semi-supervised, unsupervised, and reinforcement learning, each offering distinct advantages and applications. Supervised learning excels in prediction tasks, semi-supervised learning enhances understanding with limited labels, unsupervised learning reveals insights from unlabeled data, and reinforcement learning excels in sequential decision-making settings. Each approach uniquely contributes to machine learning challenges, catering to diverse problem domains and objectives. A comparison of these approaches, considering various attributes, is presented in Tables 6 and 7 below.

In our simulation we are using reinforcement learning with epsilon-greedy algorithm for CF. The epsilon-greedy algorithm is a commonly used strategy in reinforcement learning that balances

exploration and exploitation. In our simulation, this algorithm allows the robots to make decisions about which subtasks to pursue based on their learned Q-values while still allowing for some randomness to discover potentially better actions. The action  $a$  chosen by a robot (agent) is determined by:

$$a = \begin{cases} \text{random action with probability } \epsilon \\ \operatorname{argmax}_{a'} Q(s, a') \text{ with probability } 1-\epsilon \end{cases} \quad (1)$$

$Q(s, a')$  is the action-value function (Q-value) for state  $s$  and action  $a'$  and  $\epsilon$  is the exploration probability. After taking action  $a$  and observing a reward  $r$  from the environment, the Q-value is updated according to the Q-learning rule.

**Table 6.** Comparison between different machine learning approaches.

Factors	Supervised learning	Unsupervised learning	Semi supervised learning	Reinforcement learning
Fault tolerance	Low, sensitive to errors in the labels as it relies on labeled data for training	Low, may handle noise and outliers better as it does not require labels.	More fault tolerant by leveraging both labeled and unlabeled data.	Medium, through exploration-exploitation trade-offs
Optimal allocation	Can achieve optimal allocation by learning from labeled data and mapping inputs to correct outputs.	No, mostly aims to discover patterns and relationships in the data.	Partially, it may require more specialized approaches	Partially, it may require more specialized approaches
Scalability	High, may face challenges due to the need for labeled data and computational complexity.	High as it does not require labeled data.	High, by utilizing both labeled and unlabeled data.	Medium-high, may face challenges due to the need for labeled data and computational complexity.
Task reassignment	Difficult, not inherently designed for it, may require additional mechanisms.	Yes, it naturally clusters data into groups.	Yes, can leverage both labeled and unlabeled data to handle task reassignment.	Yes, equipped to handle task reassignment in sequential decision-making problems.
Coalition formation	Not specifically tailored, may need additional considerations and adaptations.	Same as Supervised learning	Same as Supervised learning and Unsupervised Learning	Possible in situations where agents make sequential decisions in coalition formation tasks.
Complexity	Lower, but based on the specific algorithm and techniques used within	Same as Supervised learning	Same as Supervised learning and Unsupervised Learning	Higher due to the need to learn policies for sequential decision making.



Cost	Lower for simple models but varies depending on the complexity of the model and size of the data.	Same as Supervised learning	Same as Supervised learning and Unsupervised Learning	High due to learning and exploration process.
------	---	-----------------------------	---	---

**Table 7.** Advantages and disadvantages of different machine learning approaches.

Method	Efficiency	Advantages	Disadvantages
Supervised Learning	Low-Medium	<ul style="list-style-type: none"><li>• Well-established and widely used</li><li>• Effective for classification and regression tasks</li></ul>	<ul style="list-style-type: none"><li>• Requires labeled data for training</li><li>• Limited ability to handle new or unseen data.</li></ul>
Semi-Supervised learning	Low-Medium	<ul style="list-style-type: none"><li>• Combines labeled and unlabeled data</li><li>• Can leverage both supervised and unsupervised learnings.</li></ul>	<ul style="list-style-type: none"><li>• Limited labeled data may still lead to suboptimal performance</li><li>• Sensitive to the quality of unlabeled data.</li></ul>
Unsupervised learning	Low-Medium	<ul style="list-style-type: none"><li>• Useful for clustering and data exploration tasks.</li><li>• Requires no labeled data for training</li></ul>	<ul style="list-style-type: none"><li>• May not produce interpretable results.</li><li>• Difficult in evaluating performance without ground truth</li></ul>
Reinforcement learning	Medium-High	<ul style="list-style-type: none"><li>• Suitable for sequential decision-making tasks</li><li>• Can learn from trial and error</li></ul>	<ul style="list-style-type: none"><li>• Computationally expensive</li><li>• Requires significant exploration to discover optimal policies.</li><li>• May suffer from convergence issues</li></ul>

2.5. Comparison with Different MRTA Approaches

In order to attain an optimal solution for the MRTA problem, it is imperative to take into account a range of critical factors. Effectively addressing these factors is crucial for obtaining the most favorable results in solving the MRTA problem. The comparative analysis presented in Table 8 highlights the diverse strengths and limitations of various CF methods, including behavior-based, market-based, optimization-based, and learning-based approaches. Each method exhibits distinct characteristics regarding scalability, complexity, optimality, flexibility, and robustness, influencing their applicability in real-world scenarios. Behavior-based methods, while suitable for small to moderate systems, often struggle with optimality and flexibility, primarily relying on local communication and simple task handling. Conversely, market-based methods introduce a more adaptable framework capable of accommodating complex tasks and heterogeneous robot populations, although they may encounter challenges with uncertainty and task reallocation due to their negotiation-based mechanisms.

Optimization-based approaches demonstrate superior performance in achieving optimal solutions, particularly in large systems; however, they may grapple with the complexities of multiple decision variables, impacting their robustness. Learning-based methods, particularly those utilizing reinforcement learning, emerge as a promising alternative, offering significant adaptability and improved performance through experience-based learning. Their capacity to handle dynamic

environments and complex tasks positions them as a viable option for future research endeavors in CF.

Despite their individual advantages, it is crucial to consider that these methods may not perform optimally in isolation. A hybrid approach, integrating elements from multiple methodologies, could enhance overall performance by leveraging the strengths of each while mitigating their weaknesses. For instance, combining learning-based algorithms with optimization techniques may yield robust solutions capable of adapting to changing environments and task requirements. Future studies should explore these hybrid methodologies, focusing on their scalability and efficiency in dynamic, real-world scenarios, thereby contributing to the advancement of multi-robot systems and CF strategies.

**Table 8.** Performance comparison of different MRTA strategies.

Factors	Behavior-based Methods	Market-based Method	Optimization-based Methods	Learning-based Methods
Scalability	Scalable for small to moderate size systems	Scalable for small to moderate size systems	Scalable for large systems	Can scale to large and complex systems
Complexity	Can handle simple to moderate complex tasks	Can handle complex tasks and heterogeneous robots	Can handle complex tasks and constraints	Can handle complex tasks, constraints, and heterogeneous robots
Optimality	May not always achieve optimality	Can achieve Pareto efficiency under certain conditions	Can achieve optimality under certain conditions.	Can achieve optimality under certain conditions. But guaranteed for good optimal allocation all the time.
Flexibility	Limited flexibility to adapt to new tasks or situations	Can be flexible and adaptable to changing market conditions	May be flexible depending on the optimization method used	Can be flexible and adaptable to changing environment
Robustness	May be robust to some degree of uncertainty or failures	Can be robust to some degree of market uncertainty and failures	May not be robust to uncertainty or failures.	Can improve robustness through learning from experience and failures.
Communication	Local communication among neighbour robots.	Multiple times broadcasting of winner robot details after bidding	Local communication among neighbour robots.	Local/Global communication
Objective function	Single/multiple objectives Implicit or ad-hoc	Single/multiple objectives Optimization	Single/multiple objectives Mathematical	Single/multiple objectives Learning from data

<b>Coordination type</b>	Centralized/ distributed	Centralized/ distributed	Centralized/ distributed	Decentralized
<b>task reallocation method</b>	Heuristics ruled searching/Bayesian Nash equilibrium	Iterative auctioning methods	Iterative searching and allocation	Reinforcement learning
<b>Uncertainty handling techniques</b>	Game theory/probabilistic predictive modelling	Iterative auctioning methods	Difficult to handle uncertainty	Adaptive models
<b>Constraints</b>	Can be handled in a collective manner	Difficult to conduct auctions	Complex and difficult to solve due to multiple decision variables	Varies based on learning algorithms
<b>Computational cost</b>	Higher than optimization-based strategy	Lower than optimization strategy	Higher than market-based strategy	High; need large amount of data
<b>Coalition formation</b>	Low efficiency as the approach is based on local rules without a global optimization perspective.	Moderate efficiency due to negotiation and market mechanisms	High efficiency through global optimization approaches	Moderate efficiency as it relies on learning and adaptive algorithms.
<b>Task reallocation</b>	Limited ability to perform task reallocation dynamically as it relies on predefined rules.	Efficient task reallocation due to negotiation and the market mechanism	Efficient reallocation due to optimization algorithms and centralized coordination	Adaptive due to learning algorithms and flexible decision making
<b>Collision avoidance</b>	Limited capability due to lack of sophisticated coordination mechanism	Effective collision avoidance due to price-based mechanisms and negotiations.	Effective due to optimized task allocation and coordination	Adaptive due to learning and sensor-based approaches
<b>Dynamic decision making</b>	Limited adaptability due to its rule-based and reactive characteristic	Limited adaptability as it relies on predefined market rules.	Flexible due to mathematical optimization and modeling	Flexible through adaptive learning algorithms
<b>Temporal constraints</b>	Limited support due to a lack of coordinated decision making	Moderate support due to negotiation and the market mechanism	Highly support handling temporal constraints through optimization techniques and advanced scheduling algorithms.	Highly support handling temporal constraints through learning and scheduling algorithms.

### 3. Simulation and Results

In our simulation studies, we use a Dell laptop with an Intel Core i7 processor and 16GB of RAM to run Python-based programs and simulations. The software environment includes necessary libraries such as NumPy, Matplotlib, and custom Python scripts to visualize robot movements and coalitions. The simulations do not involve real hardware robots; instead, we focus on the behavior and coalition dynamics of virtual robots modeled in the software. All robot movements and task assignments appear in real time through graphical plots generated by the Python simulation, providing clear visualization of the coalition-based task allocation process.

We consider a team of  $N$  robots, each denoted as robot  $i \in \{1, 2, 3, \dots, N\}$  and their positions are represented by  $x_i$  in the two-dimensional space  $\mathbb{R}^2$ ,  $x_i \in \mathbb{R}^2$ . Similarly, we have objects labeled as  $l \in \{1, 2, 3, \dots, M\}$  and their positions, velocities, and desired positions are represented as  $z_l \in \mathbb{R}^2$ ,  $v_l \in \mathbb{R}^2$ , and  $z_l^d \in \mathbb{R}^2$ , respectively. Robot  $i$  can observe other robots  $j$  belonging to the set  $\mathcal{N}_i$  of neighboring robots. Robot  $i$  can observe neighboring robots  $j \in \mathcal{N}_i$  and obstacle  $l \in \mathcal{L}_i$ . These observed robots and objects are selected based on their proximity to robot  $i$ , specifically, the  $K$  nearest neighbors. We have established the following assumptions for this scenario:

- Robots are aware of the values of  $M$  (the total number of objects) and  $N$  (the total number of robots).
- Robots possess knowledge of both the current and desired positions of all  $M$  objects.
- Robots are capable to communicate with each other as needed.
- We assume that all robots are operating within a workspace where communication between robots is feasible.

We simulate a scenario with a set of 50 robots, denoted as  $R = \{r_1, r_2, r_3, \dots, r_{50}\}$ , and a task,  $T = \{t_1, t_2, t_3, t_4, t_5\}$ , divided into five subtasks. The assignment of robots to these subtasks is based on their capabilities and the specific requirements of each subtask. Although robots are assumed to be identical, they differ in terms of their capabilities, categorized into three groups: high, medium, and low.

Each robot is randomly assigned to evaluate the temporal capabilities of the subtasks, and the optimal CF is determined using various algorithms. We consider the distance of each robot from each subtask and the time required for them to reach the nearest subtask. Since distance and time are directly related, each robot selects the nearest subtask to ensure efficient CF.

In this study, we evaluate the top-performing algorithms within each category. Specifically, we selected four algorithms: Alliance (behavioral-based), M+ (market-based), PSO (optimization-based), and Reinforcement Learning (learning-based). Our simulations involved running these algorithms through a series of tests using identical parameters, with each simulation consisting of 20 iterations.

After 20 iterations, we obtained 50 convergence matrices, each representing the distances of the 50 robots to the nearest subtask over the iterations. The convergence analysis assesses the speed and reliability with which the sequence of approximations, generated by the iterative method, converges toward the actual solution. The convergence matrix plays a crucial role in this analysis.

We deliberately chose 10 convergence matrices for each algorithm to capture a range of distances, spanning from the closest to the farthest robot-subtask configurations. Within these matrices, we analyzed the average distance, and the time required for individual robots to reach convergence with their respective subtasks. Subsequently, we compared the results from all four algorithms to draw conclusions from our findings. The next section will discuss the results of these algorithms.

### 3.1. Behaviour-Based: Alliance Architecture

The ALLIANCE architecture can be mathematically described in terms of robot capabilities, CF, and robot movement. Robot movement towards a subtask is governed by updating the robot's position based on the direction vector  $\vec{d}$  which points toward the subtask, scaled by the robot's velocity  $v$ . This is mathematically represented as

$$\text{New Position}_s = \text{Position}_r + \frac{\vec{d}}{|\vec{d}|} * v . (2)$$

Each robot's capability for a given subtask is denoted as  $C_{r,s} \in [0, 1]$ , where  $C_{r,s}$  represents the capability of robot  $r$  to perform subtask  $s$ . CF occurs when a robot's capability for a particular subtask is greater than zero, formally expressed as  $\text{coalition}_s = \{r | C_{r,s} > 0\}$ , where  $\text{coalition}_s$  is the set of robots suitable for subtask  $s$ .

The implementation of the ALLIANCE architecture in our simulation involves key concepts such as robot capabilities, CF, and autonomous movement. Each robot is assigned distinct capabilities for different subtasks, which are randomly generated and stored in a *robot\_capabilities* matrix. These capabilities represent the temporal suitability of each robot to perform a given subtask, based on sensor inputs or predefined characteristics.

CF is driven by these capabilities, where robots with non-zero capability values for a specific subtask form coalition. In the code, this process is executed by appending robot IDs to the respective coalitions if their capability exceeds zero.

Once the coalitions are formed, the robots autonomously navigate toward the assigned subtasks. The movement is computed by calculating the direction vector between a robot's current position and the position of the closest subtask.

$$\text{direction} = (\text{best}_{\text{subtask}_{pos[0]}} - \text{robot}_{pos[0]}, \text{best}_{\text{subtask}_{pos[1]}} - \text{robot}_{pos[1]}) (3)$$

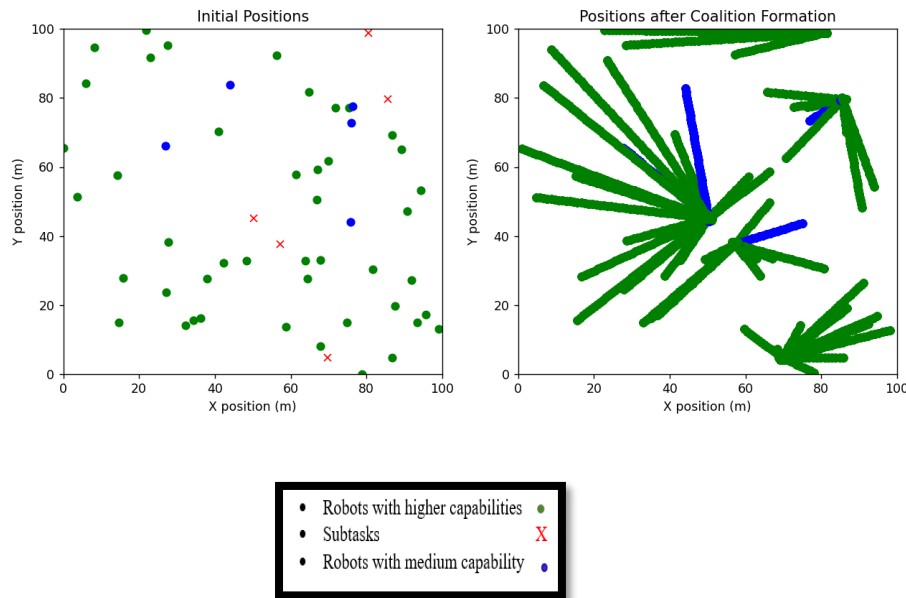
This equation computes the direction vector from a robot's current position to the position of its most suitable subtask. It does so by subtracting the robot's x and y coordinates from the corresponding coordinates of the best subtask position, selected using a function or method represented by `best_()`. The resulting vector indicates the direction the robot should move in order to approach the optimal subtask, based on certain criteria such as proximity or task relevance. This directional information is fundamental for navigation and task allocation, helping the robot align its movement towards the assigned goal efficiently.

This dynamic process ensures that each robot continuously updates its movement based on its capability and proximity to the task, reflecting the behavior-based control mechanisms inherent in the ALLIANCE architecture.

The initial simulation employed Alliance architecture with a fleet of 50 robots of differing capabilities, randomly assigned as high, medium, or low. Robots were directed towards five subtasks, selected based on proximity. Over 20 iterations, the results shown in Figure 8 illustrate efficient coalition formation (CF) using the Alliance algorithm. All 50 robots were effectively allocated to the nearest subtasks, promoting active participation and rapid task accessibility. Figure 8 displays the CF outcome after 20 iterations.

The alliance architecture facilitates the convergence of multiple robots into a single point to form a coalition through flexible and decentralized coordination. Robots autonomously organize into alliances based on shared objectives, capabilities, and task compatibility. This adaptability allows robots to dynamically identify suitable partners with aligned interests and abilities, leading to a common goal or convergence point. Consequently, the alliance architecture promotes efficient cooperation among diverse robots, enabling effective collaboration in achieving shared tasks or converging to specific points in their environment, fostering robust and scalable multi-robot systems.





**Figure 8.** Coalition formation using Alliance.

### 3.2. Market-Based: M+ Algorithm

In our simulation, the M+ algorithm is applied to create optimal CF for robots tasked with subtasks in a warehouse environment. The temporal capabilities of robots, stored in a matrix, are used to rank robots for each subtask. The top-performing robots are selected to form coalitions based on their capabilities. Coalition size is determined by a square root function, simulating optimal task distribution. Once assigned, the robots navigate toward their designated subtasks by calculating the shortest distance and updating their positions iteratively. This code implementation reflects the core mechanics of the M+ algorithm, focusing on distributed decision-making and market-based CF.

Each robot  $r$  is assigned a capability score  $C_{r,s} \in [0,1]$ , for each subtask  $s$ . This score represents the robot's ability to effectively contribute to a specific subtask, and it plays a critical role in determining the robot's eligibility to join a coalition for that subtask. In our implementation, the robot capabilities are represented by a matrix where each element is randomly generated using the following expression:

$$C_{r,s} = \text{random.uniform}(0.1, 1) \quad (4)$$

where  $C_{r,s}$  denotes the capability of robot  $r$  for subtask  $s$ , and its value is uniformly drawn from the interval  $[0.1, 1]$ , ensuring variability in robot performance across subtasks.

$$\text{coalition size} = \min(N, \lfloor \sqrt{N} \rfloor) \quad (5)$$

The algorithm selects robots with the highest capabilities to form a coalition for each subtask. The size of the coalition is determined by a function related to the total number of robots  $N$ , which approximates the optimal number of robots per coalition. The coalition size is computed as  $\lfloor \sqrt{N} \rfloor$ , reflecting the square root rule commonly used in distributed task allocation.

For each subtask  $s$ , the robots are ranked based on their capability scores  $C_{r,s}$  and the top-performing robots are selected to form the coalition:

$$\text{coalition}_s = \{r | C_{r,s} \geq \text{threshold}\} \quad (6)$$

In our code, the coalitions are formed by selecting the top  $\lfloor \sqrt{N} \rfloor$  robots from the sorted list of robots based on their capabilities.

Once the coalitions are formed, each robot in a coalition is assigned a specific subtask and moves toward the location of the subtask. The distance  $d$  between robot  $r$  and subtask  $s$  is calculated as:

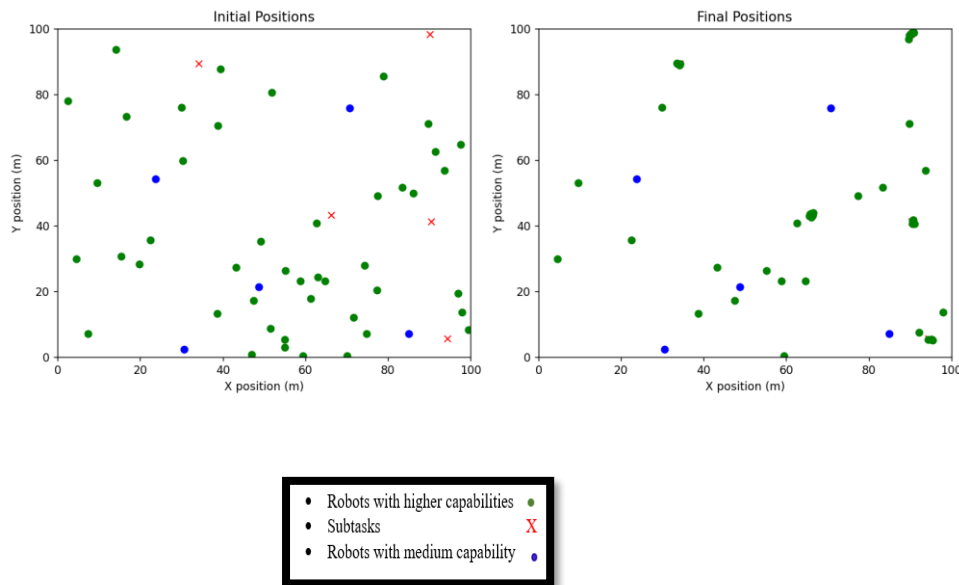
$$d = \sqrt{(x_s - x_r)^2 + (y_s - y_r)^2} \quad (7)$$

The movement of each robot is governed by minimizing the Euclidean distance between the robot's current position  $(x_r, y_r)$  and the target subtask's position  $(x_s, y_s)$ .

The algorithm iteratively updates the robot's position by moving it along the vector direction towards the subtask, ensuring that the robot gradually reduces its distance to the target subtask.

In the simulation with M+ algorithm, we employed the same 50 robots with varying capabilities, randomly assigned as high, medium, or low. These robots were tasked with moving towards 5 subtasks, with their selection based on their current proximity to the subtasks. We conducted a total of 20 iterations, and the conclusive outcome is presented in Figure 9.

The difference between alliance architecture and M+ in CF, concerning their ability to converge to a common point or task, lies in their fundamental mechanisms. Alliance architecture emphasizes a decentralized, self-organizing approach where agents form alliances based on shared interests and capabilities. Convergence in this context relies on the agents' ability to autonomously identify suitable partners and form alliances, gradually moving the system towards a desired task or point of convergence through distributed decision-making. On the other hand, M+ utilizes mathematical programming to globally optimize the allocation of tasks to agents, explicitly defining an objective function. Convergence in M+ occurs through the solution of this optimization problem, aiming for an efficient task allocation. While alliance architecture offers flexibility in dynamic environments, M+ may excel in optimizing task allocations but could be less adaptable to changing circumstances due to its reliance on a fixed optimization framework.



**Figure 9.** Coalition formation using M+.

### 3.3. Optimization-Based: PSO Algorithm

In PSO, each "particle" represents a potential solution, in our case, the position of each robot in the warehouse. The initial positions of robots are randomly generated within the dimensions of the warehouse:

$$x_i(0) = (x_i(0), y_i(0)), i = \{1, 2, 3, \dots N\} \quad (8)$$

$N$  is the number of robots, and  $X_i(0)$  represents the initial position of robot  $i$  in the 2D-space. The velocities of the particles are also initialized randomly, and the goal of the algorithm is to iteratively update these velocities and positions to optimize the assignment of robots to subtasks. The objective function that our PSO is optimizing is the ratio of each robot's capability for a subtask to its distance from that subtask. The score  $S_i$  of each particle (robot) is defined as:

$$S_i = \sum_{s=1}^M \frac{c_{i,s}}{d_{i,s}} \quad (9)$$

$c_{i,s}$  is the capability,  $d_{i,s}$  is the Euclidean distance between the position of robot  $i$  for subtask  $s$ , and  $M$  is the total number of subtasks. The score is maximized when robots with higher capabilities are closer to their respective subtasks. Each robot maintains its **personal best position**  $P_i$ , means the position where it had the highest score in the past):

$$P_i(t) = \arg \max_i S_i(t) \quad (10)$$

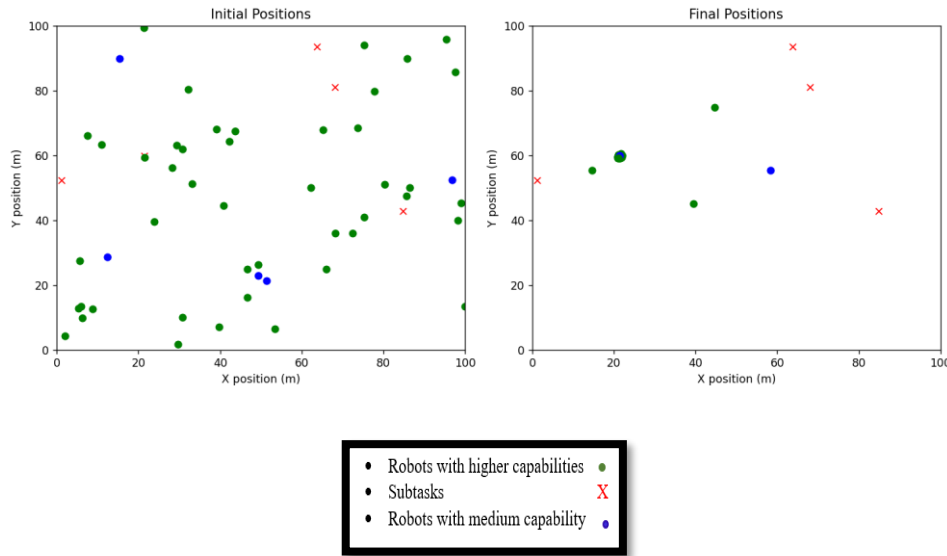
There is also a **global best position**  $g$  that corresponds to the highest score achieved by any robot across the swarm:

$$g(t) = \arg \max_t S_i(t) \quad (11)$$

In the PSO algorithm, the velocity and position of each particle, representing a robot, are updated through three key components: inertia, cognitive, and social terms. The inertia term maintains the particle's current momentum, ensuring it continues in its present direction, with the velocity update defined as  $v_i(t+1) = w * v_i(t)$ , where  $w$  represents the inertia weight. The cognitive term directs the particle towards its personal best position,  $p_i(t)$ , by adjusting its velocity based on the difference between its current position and its best-known position, represented as  $c_1 r_1 (p_i(t) - x_i(t))$ , where  $c_1$  is the cognitive weight and  $r_1$  is a random scalar between 0 and 1. The social term steers the particle towards the global best position,  $g(t)$ , using  $c_2 r_2 (g(t) - x_i(t))$ , where  $c_2$  is the social weight and  $r_2$  is a random scalar. The overall velocity update equation combines these components as  $v_i(t+1) = w * v_i(t) + c_1 r_1 (p_i(t) - x_i(t)) + c_2 r_2 (g(t) - x_i(t))$ , and the particle's position is subsequently updated using  $x_i(t+1) = x_i(t) + v_i(t+1)$ .

In this simulation, we applied the PSO algorithm to a fleet of 50 robots with diverse capabilities, randomly classified as high, medium, or low. The robots were tasked with approaching five subtasks, with selection based on their relative distances. Over 20 iterations, the results indicate that the PSO algorithm prioritized the first subtask, allocating all robots towards its completion shown in figure 10. This outcome highlights PSO's strength in converging on near-optimal solutions; however, our objective is to accelerate task completion by enabling simultaneous engagement across all subtasks to minimize total task time.

PSO differs from the Alliance Architecture and the M+ algorithm in CF, particularly in its approach to converging on a common point or task. PSO is a heuristic optimization technique inspired by the collective behavior of birds or fish in a swarm. In CF, PSO models' agents as particles exploring a solution space to find the optimal task allocation. While the Alliance Architecture and M+ emphasize decentralized coordination and explicit optimization, PSO uses a population-based search mechanism. It doesn't form traditional coalitions but directs agents (particles) towards optimal solutions. PSO's convergence is driven by the optimization process, with particles adjusting their positions based on their experiences and those of their neighbors. This stochastic exploration can effectively find near-optimal solutions, especially in continuous solution spaces, but it may lack the transparency and adaptability of the Alliance Architecture or M+ in multi-agent systems with evolving goals and capabilities.



**Figure 10.** Coalition formation using PSO.

### 3.4. Learning-Based: Reinforcement Learning

In the proposed reinforcement learning framework for multi-robot CF, the epsilon-greedy algorithm is utilized to strike a balance between exploration and exploitation during the decision-making process. Each robot maintains a Q-value table  $Q(s, a)$ , where  $s$  represents the current state of the robot, and  $a$  represents the actions associated with available subtasks. To select an action, the robot employs the epsilon-greedy strategy, wherein it explores a random action with probability  $\epsilon$  and exploits the action with the highest Q-value with probability  $1 - \epsilon$ . This dual approach facilitates the discovery of optimal strategies while ensuring that previously identified effective actions are also utilized.

The Q-value updates are performed following the Q-learning rule, allowing the robots to refine their strategies based on the rewards received from the environment. Specifically, after executing an action and observing a reward, the robot updates its Q-value for the taken action using the formula:

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)] \quad (12)$$

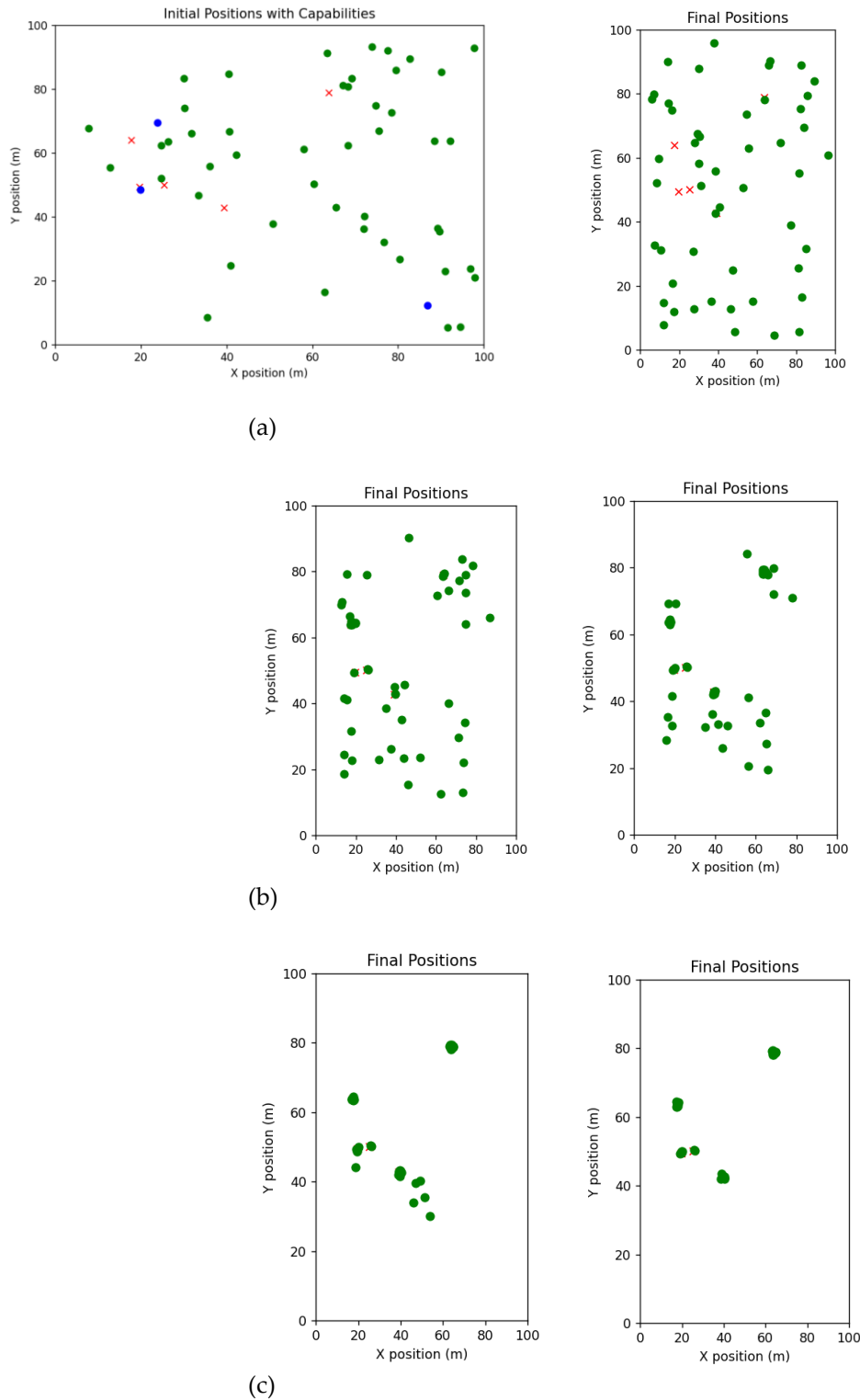
Here,  $\alpha$  represents the learning rate, which controls the extent to which new information influences the existing Q-values, while  $\gamma$  denotes the discount factor, balancing immediate versus future rewards. This method allows for an effective adaptation of the robots' actions in pursuit of subtasks, enhancing their cooperative performance in the CF process.

In our simulation employing the Reinforcement Learning (RL) algorithm, we utilized the same 50 robots with varying capabilities, randomly assigned as high, medium, or low. These robots were tasked with moving towards 5 subtasks, selecting based on their proximity to the subtasks. We conducted a set of 20 iterations, with the final results displayed in Figures 11a and 11b. A notable limitation of RL is its prolonged learning duration, hindering convergence within the initial 20 iterations. To address this, we extended the number of iterations to a range of 80 to 100, as demonstrated in Figure 11c.

RL differs significantly from alliance architecture, M+, and PSO algorithms in CF due to its agent-centric, trial-and-error learning paradigm. RL agents make decisions based on learned policies, and convergence occurs through individual agent learning. While alliance architecture and M+ emphasize decentralized coordination and explicit CF, RL agents act autonomously to maximize cumulative rewards, often without explicit coalition considerations. Conversely, PSO aims to converge on optimal solutions collectively. RL introduces adaptability and autonomy at the individual agent level, allowing agents to dynamically adapt to changing circumstances and goals.

The choice between these approaches depends on factors such as problem complexity, collaboration requirements, desired autonomy and adaptability in the multi-robot system.

From the results depicted in Figure 11, it is evident that RL exhibits behavior similar to alliance architecture, indicating its significant value.



**Figure 11.** CF using RL: a) initial positions b) CF using RL in 20 iterations c) CF using RL in 100 iterations.

### 3.5. Statistical Analysis

Figure 12 depicts behavioral bar charts for a set of 10 chosen robots. These charts illustrate the utilization of alliance, M+, PSO, and reinforcement learning algorithms. In the charts, we can observe

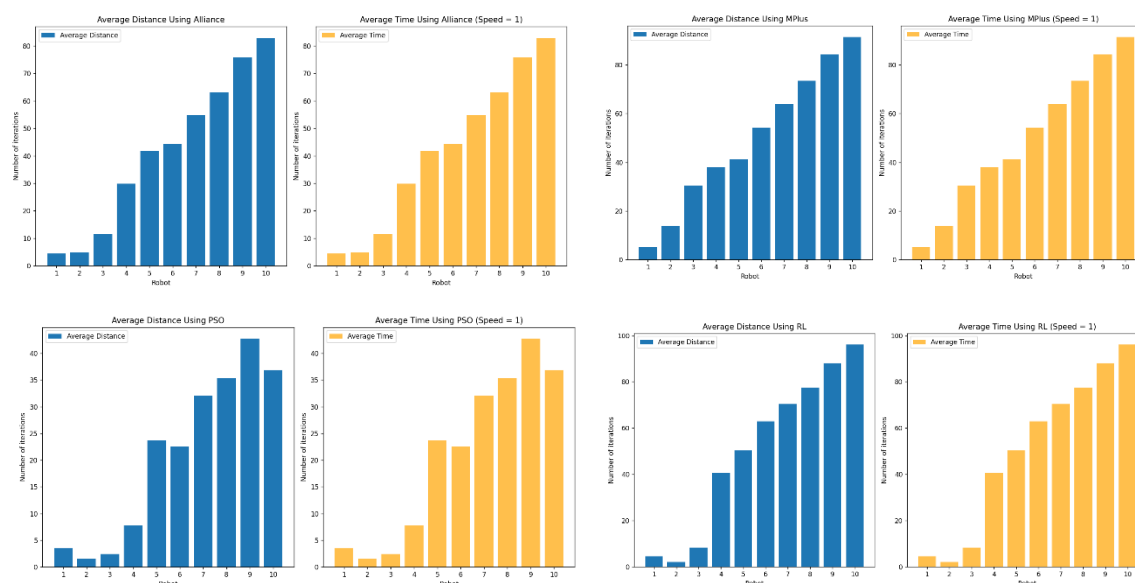


blue bars representing the average distance covered by each robot, while the yellow bars represent the average time required for convergence. Notably, the starting position of R1 is in proximity to the subtask, whereas R10's initial position is considerably distant from the subtask.

The results make it evident that robots employing alliance and reinforcement learning algorithms exhibit superior performance. However, it's worth noting that the reinforcement learning algorithm, due to its learning attributes, takes more time to accomplish the task compared to the alliance approach. Yet, as the number of robots increases, the performance of the alliance architecture starts to deteriorate, and this is when reinforcement learning shines. If we can mitigate the time constraints associated with reinforcement learning, it would emerge as the optimal approach for CF and multi-robot task allocation.

In the analysis of CF algorithms, the four approaches, Alliance (behavior-based), M+ (market-based), PSO (optimization-based), and reinforcement learning (Learning-based), exhibit distinct methodologies and effectiveness in promoting collaborative behavior among autonomous agents. The Alliance algorithm emphasizes behavioral dynamics, enabling robots to form coalitions based on mutual capabilities and situational awareness. In contrast, the M+ algorithm leverages market mechanisms to allocate tasks, optimizing efficiency through competitive bidding among agents. The PSO approach, rooted in swarm intelligence, focuses on optimizing coalition structures based on collective experience and positioning, promoting swift convergence towards optimal solutions. While both the Alliance and RL methods yield promising results in CF, the reinforcement learning approach emerges as a robust framework for future studies. This is primarily due to its adaptive learning capabilities, which allow robots to continuously improve their decision-making processes through trial and error, leading to enhanced performance in dynamic environments.

Despite the preliminary findings indicating effective results from the Alliance and RL algorithms, the inclusion of reinforcement learning is particularly compelling for future investigations. It offers the potential for self-optimization and adaptability, accommodating varying task demands and environmental conditions that may not be fully captured by the other methods. Moreover, the comparative analysis conducted through convergence matrices of 10 robots showcases the efficiency of these algorithms, highlighting RL's capability to converge toward optimal coalition structures over time. Such insights pave the way for further exploration of RL in CF, potentially leading to more resilient and flexible robotic systems capable of adapting to complex and evolving scenarios.



**Figure 12.** Convergence Metrics results for 10 Robots with Alliance, M+, PSO, and RL.

## 4. Discussion

Upon reviewing the literature, it becomes evident that learning-based methods hold the potential to adapt and attain a truly optimal solution for the Multi-Robot Task Allocation (MRTA) problem. Each of these methods, including behavior-based, market-based, and optimization-based approaches, may excel in certain aspects, while others might fall short. Nevertheless, learning-based approaches, when applied to the MRTA problem, have the capability to offer solutions, even when dealing with large teams. When analyzing algorithms against mTSP, VRP, JSP, TOP and DARP; they are not scaled well to many robots or locations, and they may struggle with real-time adaptation to dynamic environments except learning-based algorithms.

Learning-based MRTA approaches are adept at tackling exceedingly intricate problems that conventional techniques struggle to address. Previous research has underscored the competitive nature of reinforcement learning in achieving long-term outcomes in this context. Furthermore, it possesses the capability to rectify errors that may have arisen during the training phase and can adapt when confronted with an insufficient quantity of training data, drawing insights from its experiences. Notably, one of the primary strengths of reinforcement learning lies in its capacity to strike a balance between exploration and exploitation. Exploration involves the testing of novel ideas to discover potential superior solutions, while exploitation entails leveraging the strategies that have previously proven to be effective. The primary challenge associated with learning-based Multi-Robot Task Allocation (MRTA) is the time required for training. Achieving a high number of reward points necessitates a substantial number of trials, which can be time intensive.

Interestingly, an intriguing avenue for addressing the limitations of learning based MRTA is to combine these approaches with one or more alternative methods that can mitigate their drawbacks. Such hybrid approaches hold the promise of yielding viable solutions to the MRTA problem. Additionally, more intricate learning techniques, such as deep reinforcement learning or deep neural networks, have the potential to enhance performance. Moreover, there exists ample room for future research and development in the realm of learning based MRTA, offering exciting prospects for further advancements.

The future research will primarily focus on addressing these challenges, particularly partial observability, and non-stationarity in CF, using a learning-based approach.

## 5. Conclusion

CF plays a critical role in enhancing the efficiency and effectiveness of MRTA systems. Through an in-depth examination of existing strategies, we highlight that while various methods such as behaviour-based, market-based, optimization-based approaches and learning based offer viable solutions, each comes with its own advantages and limitations. The comparative analysis emphasizes that no single method fits all scenarios; rather, the effectiveness of a CF strategy is heavily dependent on the nature of the task, the capabilities of the robots, and the dynamic environment in which they operate. Our simulation results affirm that strategic grouping of robots based on proximity, capability, and task requirements leads to more reliable and faster convergence toward optimal task execution.

In conclusion, this paper not only provides a structured overview of current MRTA approaches with a focus on CF but also proposes a classification system to better understand and compare these methodologies. The introduced simulation-based evaluation framework further substantiates the potential of the proposed CF approach in solving complex, real-world MRTA problems. Future work may explore hybrid models that dynamically switch between strategies based on environmental changes and robot feedback, paving the way for more adaptive, scalable, and intelligent multi-robot systems.

## References

1. Gautam, A. and S. Mohan. *A review of research in multi-robot systems*. in 2012 IEEE 7th International Conference on Industrial and Information Systems (ICIIS). 2012.

2. Parker, L., *Distributed intelligence: overview of the field and its application in multi-robot systems*. Journal of Physical Agents, 2008. **2**.
3. Arai, T. and L. Parker, *Editorial: advances in multi-robot systems*. 2003.
4. Roldán Gómez, J. and A. Barrientos, *Special issue on multi-robot systems: challenges, trends, and applications*. Applied Sciences, 2021. **11**: p. 11861.
5. Khamis, A., A. Hussein, and A. Elmogy, *Multi-robot task allocation: a review of the state-of-the-art*. 2015. p. 31-51.
6. Gerkey, B.P. and M.J. Mataric. *Multi-robot task allocation: analyzing the complexity and optimality of key architectures*. in 2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422). 2003.
7. Gerkey, B. and M. Mataric, *A formal analysis and taxonomy of task allocation in multi-robot systems*. I. J. Robotic Res., 2004. **23**: p. 939-954.
8. Korsah, G., A. Stentz, and M. Dias, *A comprehensive taxonomy for multi-robot task allocation*. International Journal of Robotics Research, 2013. **32**: p. 1495-1512.
9. Saravanan, S., K.C. Ramanathan, R. Mm, and M.N. Janardhanan, *Review on state-of-the-art dynamic task allocation strategies for multiple-robot systems*. Industrial Robot, 2020.
10. Wen, X. and Z.G. Zhao. *Multi-robot task allocation based on combinatorial auction*. in 2021 9th International Conference on Control, Mechatronics and Automation (ICCMA). 2021.
11. Chakraa, H., F. Guérin, E. Leclercq, and D. Lefebvre, *Optimization techniques for multi-robot task allocation problems: review on the state-of-the-art*. Robotics and Autonomous Systems, 2023. **168**: p. 104492.
12. Dos Reis, W.P.N., G.L. Lopes, and G.S. Bastos, *An arrowian analysis on the multi-robot task allocation problem: analyzing a behavior-based architecture*. Robotics and Autonomous Systems, 2021. **144**: p. 103839.
13. Agrawal, A., A. Bedi, and D. Manocha, *RTAW: an attention inspired reinforcement learning method for multi-robot task allocation in warehouse environments*. 2023. 1393-1399.
14. Cheikhrouhou, O. and I. Khoufi, *A comprehensive survey on the multiple traveling salesman problem: applications, approaches and taxonomy*. Computer Science Review, 2021. **40**: p. 100369.
15. Deng P, Amirjamshidi G, Roorda M. *A vehicle routing problem with movement synchronization of drones, sidewalk robots, or foot-walkers*. Transportation Research Procedia, 2020. **46**: p. 29-36.
16. Sun Y, Chung S-H, Wen X, Ma H-L. *Novel robotic job-shop scheduling models with deadlock and robot movement considerations*. Transportation Research Part E: Logistics and Transportation Review, 2021. **149**: p. 102273.
17. Santana KA, Pinto VP, Souza DAd, Torres JLO, Teles IAG, editors. *New GA applied route calculation for multiple robots with energy restrictions*. 2020.
18. Jorgensen R, Larsen J, Bergvinsdottir K. *Solving the dial-a-ride problem using genetic algorithms*. Journal of the Operational Research Society, 2007. **58**.
19. Hussein, A. and A. Khamis. *Market-based approach to multi-robot task allocation*. in 2013 International Conference on Individual and Collective Behaviors in Robotics (ICBR). 2013.
20. Ramanathan, K.C., M. Singaperumal, and T. Nagarajan, *Cooperative formation planning and control of multiple mobile robots*. 2011.
21. Aziz, H., et al., *Task allocation using a team of robots*. Current Robotics Reports, 2022. **3**.
22. Zitouni, F., S. Harous, and R. Maamri, *A distributed approach to the multi-robot task allocation problem using the consensus-based bundle algorithm and ant colony system*. IEEE Access, 2020. **8**: p. 27479-27494.
23. Rauniyar, A. and P.K. Muhuri. *Multi-robot coalition formation problem: task allocation with adaptive immigrants based genetic algorithms*. in 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC). 2016.
24. Kraus, S., *Negotiation and cooperation in multi-agent environments*. Artificial Intelligence, 1997. **94**(1): p. 79-97.
25. Tasic, P. and C. Ordonez, *Distributed protocols for multi-agent coalition formation: a negotiation perspective*. 2012.
26. Selseleh Jonban, M., A. Akbarimajd, and M. Hassanpour, *A combinatorial auction algorithm for a multi-robot transportation problem*. 2014.
27. Capitan J, Spaan MTJ, Merino L, Ollero A. *Decentralized multi-robot cooperation with auctioned POMDPs*. The International Journal of Robotics Research. 2013;32(6): p. 650-71.

28. Hernandez-Leal P, Kaisers M, Baarslag T, Munoz de Cote E. *A survey of learning in multiagent environments: dealing with non-stationarity*. 2017.
29. Vig, L. and J.A. Adams, *Multi-robot coalition formation*. IEEE Transactions on Robotics, 2006. **22**(4): p. 637-649.
30. Guerrero, J. and G. Oliver, *Multi-robot coalition formation in real-time scenarios*. Robotics and Autonomous Systems, 2012. **60**(10): p. 1295-1307.
31. Rizk, Y., M. Awad, and E. Tunstel, *Cooperative heterogeneous multi-robot systems: A Survey*. 2019.
32. Ramanathan, K.C., M. Singaperumal, and T. Nagarajan, *Behaviour based planning and control of leader follower formations in wheeled mobile robots*. International Journal of Advanced Mechatronic Systems, 2010. **2**.
33. Schillinger, P., M. Bürger, and D. Dimarogonas, *Simultaneous task allocation and planning for temporal logic goals in heterogeneous multi-robot systems*. The International Journal of Robotics Research, 2018. **37**(7): p. 818-838.
34. Parker, L.E., *ALLIANCE: an architecture for fault tolerant multirobot cooperation*. IEEE Transactions on Robotics and Automation, 1998. **14**(2): p. 220-240.
35. Parker, L.E. *ALLIANCE: an architecture for fault tolerant, cooperative control of heterogeneous mobile robots*. in Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94). 1994.
36. Parker, L.E., *L-ALLIANCE: task-oriented multi-robot learning in behavior-based systems*. Advanced Robotics, 1996. **11**(4): p. 305-322.
37. Parker, L., *L-ALLIANCE: a mechanism for adaptive action selection in heterogeneous multi-robot teams*. 1996.
38. Parker, L., *Evaluating success in autonomous multi-robot teams: experiences from ALLIANCE architecture implementations*. Journal of Experimental & Theoretical Artificial Intelligence, 2000. **13**.
39. Parker, L., *On the design of behavior-based multi-robot teams*. Advanced Robotics, 2002. **10**.
40. Parker, L.E. *Task-oriented multi-robot learning in behavior-based systems*. in Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS '96. 1996.
41. Mesterton-Gibbons, M., S. Gavrillets, J. Gravner, and E. Akçay, *Models of coalition or alliance formation*. Journal of theoretical biology, 2011. **274**: p. 187-204.
42. Dahl, T., M. Matarić, and G. Sukhatme, *Multi-robot task allocation through vacancy chain scheduling*. Robotics and Autonomous Systems, 2009. **57**: p. 674-687.
43. Lerman, K., A. Galstyan, A. Martinoli, and A.J. Ijspeert, *A macroscopic analytical model of collaboration in distributed robotic systems*. Artificial life, 2001. **7**: p. 375-93.
44. Jia, X. and M.Q.H. Meng. *A survey and analysis of task allocation algorithms in multi-robot systems*. in 2013 IEEE International Conference on Robotics and Biomimetics (ROBIO). 2013.
45. Werger, B. and M. Mataric, *Broadcast of local eligibility: behavior-based control for strongly cooperative robot teams*. 2000.
46. Werger, B. and M. Mataric, *Broadcast of local eligibility for multi-target observation*. 2000. p. 347-356.
47. Faigl, J., M. Kulich, and L. Preucil, *Goal assignment using distance cost in multi-robot exploration*. 2012. p. 3741-3746.
48. Tang, F. and L. Parker, *ASyMTRe: automated synthesis of multi-robot task solutions through software reconfiguration*. 2005. p. 1501-1508.
49. Fang, T. and L.E. Parker. *Coalescent multi-robot teaming through ASyMTRe: a formal analysis*. in ICAR '05. Proceedings., 12th International Conference on Advanced Robotics, 2005.
50. Fang, G., G. Dissanayake, and H. Lau, *A behaviour-based optimisation strategy for multi-robot exploration*. 2005. **2**: p. 875-879.
51. Trigui, S., et al., *A distributed market-based algorithm for the multi-robot assignment problem*. Procedia Computer Science, 2014. **32**: p. 1108-1114.
52. Badreldin, M., A. Hussein, and A. Khamis, *A comparative study between optimization and market-based approaches to multi-robot task allocation*. Advances in Artificial Intelligence, 2013. p. 11.
53. Service, T.C., S.D. Sen, and J.A. Adams. *A simultaneous descending auction for task allocation*. in 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC). 2014.
54. Vig, L. and J. Adams, *A framework for multi-robot coalition formation*. 2005. p. 347-363.
55. Vig, L. and J. Adams, *Market-based multi-robot coalition formation*. 2007. p. 227-236.

56. Vig, L. and J.A. Adams, *Coalition formation: from software agents to robots*. Journal of Intelligent and Robotic Systems, 2007. **50**(1): p. 85-118.
57. Lueth, T. and T. Längle, *Task description, decomposition, and allocation in a distributed autonomous multi-agent robot system*. 1994.
58. Längle, T., T. Lueth, and U. Rembold, *A distributed control architecture for autonomous robot systems*. 1994. p. 384-402.
59. Gerkey, B.P. and M.J. Mataric, *Sold!: auction methods for multirobot coordination*. IEEE Transactions on Robotics and Automation, 2002. **18**(5): p. 758-768.
60. Guidotti, C.F., A.T. Baião, G.S. Bastos, and A.H.R. Leite. *A murdoch-based ROS package for multi-robot task allocation*. in 2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE). 2018.
61. Lagoudakis, M., et al., *Auction-based multi-robot routing*. 2005. p. 343-350.
62. Liu, L. and Z. Zhiqiang. *Combinatorial bids based multi-robot task allocation method*. in Proceedings of the 2005 IEEE International Conference on Robotics and Automation. 2005.
63. Sheng, W., Q. Yang, J. Tan, and N. Xi, *Distributed multi-robot coordination in area exploration*. Robotics and Autonomous Systems, 2006. **54**(12): p. 945-955.
64. Gerkey, B. and M. Mataric, *MURDOCH: publish/subscribe task allocation for heterogeneous agents*. 2000.
65. Alami, R., et al., *Multi-robot cooperation in the MARTHA project*. IEEE Robotics Autom. Mag., 1998. **5**: p. 36-47.
66. Botelho, S. and R. Alami, *M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement*. 1999. 2: p. 1234-1239.
67. Botelho, S.C. and R. Alami. *A multi-robot cooperative task achievement system*. in Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065). 2000.
68. Smith, *The contract net protocol: high-level communication and control in a distributed problem solver*. IEEE Transactions on Computers, 1980. **C-29**(12): p. 1104-1113.
69. Dias, M. B., Zlot, R., Zinck, M., Gonzalez, J. P., & Stentz, A. (2018, June 29). *A versatile implementation of the traderbots approach for multirobot coordination*. Carnegie Mellon University. <https://doi.org/10.1184/R1/6551405.v1>
70. Dias, M.B. and A. Stentz. *A free market architecture for distributed control of a multirobot system*. 2000.
71. Zlot, R., A. Stentz, M.B. Dias, and S. Thayer. *Multi-robot exploration controlled by a market economy*. in Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292). 2002.
72. Dias, M.B. and A. Stentz, *Traderbots: a new paradigm for robust and efficient multirobot coordination in dynamic environments*. 2004, Carnegie Mellon University.
73. Hussein, A., P. Marín-Plaza, F. García, and J.M. Armingol, *Hybrid optimization-based approach for multiple intelligent vehicles requests allocation*. Journal of Advanced Transportation, 2018. p. 2493401.
74. Shelkamy, M., C.M. Elias, D.M. Mahfouz, and O.M. Shehata. *Comparative analysis of various optimization techniques for solving multi-robot task allocation problem*. in 2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES). 2020.
75. Atay, N. and B. Bayazit, *Mixed-integer linear programming solution to multi-robot task allocation problem*. 2006.
76. Bouyarmane, K., J. Vaillant, K. Chappellet, and A. Kheddar, *Multi-robot and task-space force control with quadratic programming*. 2017.
77. Pugh, J. and A. Martinoli. *Inspiring and modeling multi-robot search with particle swarm optimization*. in 2007 IEEE Swarm Intelligence Symposium. 2007.
78. Imran, M., R. Hashim, and N.E.A. Khalid, *An overview of particle swarm optimization variants*. Procedia Engineering, 2013. **53**: p. 491-496.
79. Li, X. and H.x. Ma. *Particle swarm optimization based multi-robot task allocation using wireless sensor network*. in 2008 International Conference on Information and Automation. 2008.
80. Nedjah, N., R. Mendonça, and L. Mourelle, *PSO-based distributed algorithm for dynamic task allocation in a robotic swarm*. Procedia Computer Science, 2015. **51**: p. 326-335.



81. Pendharkar, P.C., *An ant colony optimization heuristic for constrained task allocation problem*. Journal of Computational Science, 2015. 7: p. 37-47.
82. Dorigo, M., M. Birattari, and T. Stützle, *Ant colony optimization: artificial ants as a computational intelligence technique*. IEEE Computational Intelligence Magazine, 2006. 1: p. 28-39.
83. Agarwal, M., et al., *Parallel multi-objective multi-robot coalition formation*. Expert Systems with Applications, 2015. 42.
84. Dorigo, M. and G.D. Caro. *Ant colony optimization: a new meta-heuristic*. in Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406). 1999.
85. Wang, J., Y. Gu, and X. Li, *Multi-robot task allocation based on ant colony algorithm*. Journal of Computers, 2012. 7.
86. Jianping, C., Y. Yimin, and W. Yunbiao, *Multi-robot task allocation based on robotic utility value and genetic algorithm*. 2009. p. 256-260.
87. Rauniyar, A. and P. Muhuri, *Multi-robot coalition formation problem: task allocation with adaptive immigrants based genetic algorithms*. 2016. p. 137-142.
88. Haghi Kashani, M. and M. Jahanshahi, *Using simulated annealing for task scheduling in distributed systems*. Computational Intelligence, Modelling and Simulation, International Conference on, 2009. p. 265-269.
89. Mosteo, A. and L. Montano, *Simulated annealing for multi-robot hierarchical task allocation with flexible constraints and objective functions*. 2006.
90. Chakraborty, S. and S. Bhowmik, *Job shop scheduling using simulated annealing*. 2013.
91. Elfakharany, A., R. Yusof, and Z. Ismail, *Towards multi robot task allocation and navigation using deep reinforcement learning*. Journal of Physics: Conference Series, 2020. 1447: p. 012045.
92. Dahl, T., M. Mataric, and G. Sukhatme, *A machine learning method for improving task allocation in distributed multi-robot transportation*. 2004.
93. Wang, Y. and C. Silva, *A machine-learning approach to multi-robot coordination*. Engineering Applications of Artificial Intelligence, 2008. 21: p. 470-484.
94. Cunningham, P., M. Cord, and S. Delany, *Supervised learning*. 2008. p. 21-49.
95. Sermanet, P., C. Lynch, J. Hsu, and S. Levine. *Time-contrastive networks: self-supervised learning from multi-view observation*. in 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). 2017.
96. Teichman, A. and S. Thrun, *Tracking-based semi-supervised learning*. The International Journal of Robotics Research, 2012. 31(7): p. 804-818.
97. Xu, J., S. Zhu, H. Guo, and S. Wu, *Automated labeling for robotic autonomous navigation through multi-sensory semi-supervised learning on big data*. IEEE Transactions on Big Data, 2021. 7(1): p. 93-101.
98. Bousquet, O., U. Luxburg, and G. Rätsch, *Advanced lectures on machine learning*. ML Summer Schools 2003, Canberra, Australia, February 2-14, 2003, Tübingen, Germany, August 4-16, 2003, Revised Lectures. 2004.
99. Arel, I., C. Liu, T. Urbanik, and A. Kohls, *Reinforcement learning-based multi-agent system for network traffic signal control*. Intelligent Transport Systems, IET, 2010. 4: p. 128-135.
100. Verma, J. and V. Ranga, *Multi-robot coordination analysis, taxonomy, challenges and future scope*. Journal of Intelligent & Robotic Systems, 2021. 102.
101. Wang, Y. and C.W.D. Silva. *Multi-robot box-pushing: single-agent Q-learning vs. team Q-learning*. in 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems. 2006.
102. Guo, H. and Y. Meng. *Dynamic correlation matrix based multi-Q learning for a multi-robot system*. in 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems. 2008.
103. Ahmad, A., et al. *Autonomous aerial swarming in GNSS-denied environments with high obstacle density*. in 2021 IEEE International Conference on Robotics and Automation (ICRA). 2021. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9561284>.
104. **Team CERBERUS wins the DARPA Subterranean Challenge. Autonomous robots lab.** <https://www.autonomousrobotslab.com/>
105. Autonomous Mobile Robots (AMR) For Factory Floors: Key Driving Factors.2021. roboticsbiz. <https://roboticsbiz.com/autonomous-mobile-robots-amr-for-factory-floors-key-driving-factors/>



106. Different Types Of Robots Transforming The Construction Industry.2020. roboticsbiz.  
<https://roboticsbiz.com/different-types-of-robots-transforming-the-construction-industry/>
107. Robocup soccer small-size league. Robocup. <https://robocupthailand.org/services/robocup-soccer-small-size-league/>
108. Robots in agriculture and farming. 2022. Cyber-weld robotic system integrators.  
<https://www.cyberweld.co.uk/robots-in-agriculture-and-farming>
109. Arjun, K. Parlevliet, D. Wang, H. Yazdani, A. Analyzing multi-robot task allocation and coalition formation methods: A comparative study. in 2024 International Conference on Advanced Robotics, Control, and Artificial Intelligence. 2024.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.