Extending Consensus-based Task Allocation Algorithms with Bid Intercession to Foster Mixed-Initiative

Victor Guillet
DTIS, ONERA, Université de Toulouse
Toulouse, France
victor.guillet@onera.fr

Christophe Grand
DTIS, ONERA, Université de Toulouse
Toulouse, France
christophe.grand@onera.fr

Charles Lesire
DTIS, ONERA, Université de Toulouse
Toulouse, France
charles.lesire@onera.fr

Gauthier Picard
DTIS, ONERA, Université de Toulouse
Toulouse, France
gauthier.picard@onera.fr

ABSTRACT

We propose a new approach for controlling task allocation in teams of robots with different capabilities. This approach allows human operators, who have a better understanding of the situation, to influence or even dictate how tasks are distributed, whilst allowing autonomous decisions. Our method works within existing consensus-based allocation algorithms by introducing *intercession* in the bidding process. Intercession allows agents to bid on behalf of others. This allows for a flexible range of control, from completely decentralized to fully human-controlled, without refactoring the consensus-based allocation scheme, which has been proven to be efficient. We build upon an existing algorithm, Consensus-based Bundle Auction (CBBA), while maintaining its solution quality and ability to reach agreement (convergence). We test our new method, I-CBBA, in simulated multi-robot task allocation (MRTA) scenarios using the ROS framework.

CCS CONCEPTS

• **Computing methodologies → Multi-agent systems**; **Cooperation and coordination**; **Robotic planning**.

KEYWORDS

Task Allocation; Bid Intercession; Mixed-Initiative; Distributed Robot Systems; Multi-Robot Systems; Auctions; Consensus

ACM Reference Format:
Victor Guillet, Christophe Grand, Charles Lesire, and Gauthier Picard. 2025. Extending Consensus-based Task Allocation Algorithms with Bid Intercession to Foster Mixed-Initiative. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 9 pages.

1 INTRODUCTION

Multi-robot systems (MRS) have emerged as a promising technology with diverse applications in sectors like search and rescue, environmental monitoring, and infrastructure inspection. Their ability to perform tasks autonomously and efficiently renders them highly valuable. Multi-robot Task Allocation (MRTA) addresses the critical challenge of assigning a set of tasks to a team of robots with varying capabilities [5]. The primary objective of MRTA is to optimize system performance, typically measured by minimizing task completion time or maximizing task coverage, while considering various operational constraints such as robot capabilities, task dependencies, and environmental factors. MRTA encompasses a spectrum of techniques, ranging from centralized algorithms with global decision-making to distributed approaches where robots collaborate to determine task assignments [8, 14]. The successful development and implementation of MRTA algorithms are crucial for facilitating effective teamwork within MRS, ultimately unlocking their full potential across various application domains.

Achieving optimal performance in MRTA scenarios often requires a delicate equilibrium between automated task allocation and human oversight [6]. While automated algorithms have demonstrably achieved impressive results, concerns persist regarding the relinquishing of complete control to centralized or decentralized decision-making processes, such as auction-based methods [7] and consensus-based approaches [3]. This aligns with the limitations identified in mixed-initiative control frameworks [1, 2]. Moreover, human operators, equipped with domain expertise and real-time situational awareness, could offer invaluable insights and adapt to unforeseen circumstances that may challenge purely algorithmic approaches [15]. Consensus-based task allocation is a decentralized approach for MRS where robots collaborate to determine task assignments. Unlike centralized methods with a single decision-maker, each robot in a consensus-based system considers its capabilities, task dependencies, and environmental factors to propose its own allocation plan. Through iteration, robots reach an agreement on an optimal task distribution. This approach leverages the collective intelligence of the robot team while promoting robustness and scalability in complex environments.

This work tackles the challenge of integrating human control within the decentralized MRTA framework. We propose a novel concept: *intercession* in consensus-based auction mechanisms. This approach endeavors to achieve a critical equilibrium between algorithmic efficiency and human oversight. By introducing intercession, human operators gain enhanced control over task allocation without sacrificing the robustness, flexibility and performance of automated auction-based allocation methods.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

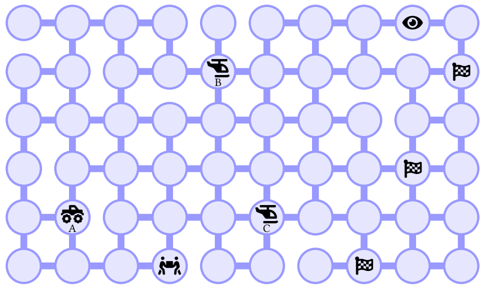


Figure 1: A sample S&R scenario. and represent the ground robot and the drones, the goto tasks, and action tasks for the rest (observation and resupply).

This paper is structured as follows. Section 2 presents the scenario motivating our contributions. Section 3 provides background on MRTA and consensus-based task allocation, along with related works. Section 4 expounds the core contribution of the paper, I-CBBA, an extension to CBBA [3] with intercession. We evaluate I-CBBA on synthetic scenarios using a ROS implementation in Section 5. Section 6 concludes the paper with some perspectives.

2 MOTIVATING MULTI-ROBOT SCENARIO

We consider a Search and Rescue (S&R) scenario, illustrated in Figure 1, where teams of autonomous robots, such as drones or rovers, are deployed to locate and assist survivors in disaster areas, which can include sinister such as fires and building collapses [11]. Equipped with sensors, cameras, and specialized tools, these robots navigate through rubble, hazardous terrain, and other challenging conditions to search for signs of life. The goal is to quickly locate survivors, assess their condition, and coordinate rescue efforts to address the various situations as efficiently and effectively as possible. The missions may therefore generally be broken down into two distinct sets of tasks: the *exploration tasks*, where robots are sent to positions to explore as to detect survivors and situations to address (which we will refer to as goto tasks), and the *action tasks*, such as rescuing victims, clearing debris, and extinguishing fires (which we will refer to as action tasks, for instance, actions a_1 and a_2). Given the heterogeneous nature of the challenges handled, agents are therefore usually specialized, and each responsible of addressing different subsets based on task requirements. Once an action task is identified, the robots autonomously allocate these tasks among themselves based on distance and their respective skills.

It is important to note that the task allocation process exclusively considers factors explicitly modeled, such as distance and specialization. It consequently overlooks non-captured factors like environmental changes or unexpected obstacles/situations. Incorporating more factors into the situation analysis and evaluation increases computational complexity, requiring sophisticated models and extensive data processing, which can be computationally intensive and harder to acquire and maintain. This complexity also poses scalability issues, potentially leading to delays in decision-making as the number of robots and tasks increases.

Consequently, *mixed-initiative systems* prove particularly valuable as human operators possess contextual knowledge or expertise that cannot be fully encapsulated by algorithms alone. Operators may have insights into survivor behavior, environmental hazards, or the significance of certain clues that might not be apparent to autonomous systems. For instance, an operator noticing smoke from a distance can infer the presence of fire and redirect a firefighting robot to the location, preventing delays and optimizing resource use through the interpretation of subtle environmental cues. Additionally, operator intervention enhances safety by addressing unforeseen hazards that robots might not detect, such as rerouting a robot away from structurally unstable areas to protect both the machines and nearby individuals. This proactive approach ensures safer operations in dynamic, unpredictable environments. Furthermore, human oversight adds reliability and robustness to multi-robot systems by allowing intervention during failures –such as sensor malfunctions– so that an operator can manually reassign tasks, ensuring the mission continues efficiently even when autonomous systems face issues.

In consequence, our approach aims to leverage the respective strength of autonomous systems in efficient problem-solving, and the human operators in observation and interpretation, to obtain a practical solution in an uncontrolled setting of the likes of S&R by introducing a generic intervention method to influence and control an autonomous allocation process when desired.

The following *driving scenario* is therefore proposed to help illustrate the challenge addressed, the limitations of the existing methods, and possible applications of our solution:

Example 1 (Driving scenario). *In a S&R scenario following an earthquake, a fleet of 3 robots (*A *and* B*, and* C*) is tasked with delivering medical supplies (symbolized by) to survivors at a specific location (task* T*):* A *is a ground robot , and* B *and* C *are drones . The robots seek to distribute tasks among themselves through an auction process.* A *wins the task with a bid* $b_A = 10$*, calculated based on proximity to the task location, followed by* C *with* $b_C = 8$ *and* B *with* $b_B = 6$*. However, an operator* O *wishes to intervene based on additional situational knowledge: The operator knows the building where survivors are located is unstable due to aftershocks. While* A *is closer, its heavier weight could exacerbate the instability. Additionally, the operator also identifies a downed power line on the ground (*A*'s route), which the robots' sensors missed.*

3 BACKGROUND

This section identifies the core problem and related consensus-based solution methods from the literature, before discussing works related to human participation in such settings.

3.1 Multi-Robot Task Allocation Problem

The multi-robot assignment problem, also known as the Multi-Robot Task Allocation [5] (MRTA) problem, refers to the challenge of assigning N_t tasks to N_u agents, to obtain a conflict-free distribution of tasks to agents that maximize some overall reward (or minimize some overall cost). An allocation is qualified as "conflict-free" if each distinct task is assigned to at most one agent. A maximum of L_t tasks can be assigned to each agent, and the assignment is considered as completed once $N_{\min} \triangleq \min\{N_t, N_u L_t\}$ tasks have been assigned. This problem has been extensively studied and for the sake of maintaining consistency with the existing literature,

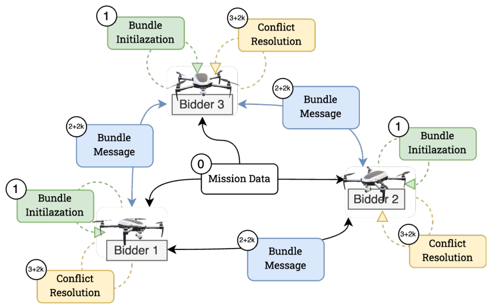


Figure 2: Consensus-based Task Allocation: agents alternate between bundle construction (auction) and conflict resolution (consensus) (from [14], with permission).

the following integer (possibly non-linear) program formulation proposed in [3] will be adopted to formalize the problem.

$$\max \quad \sum_{i=1}^{N_u}\left(\sum_{j=1}^{N_t} c_{ij}\left(\mathbf{x}_i, \mathbf{p}_i\right) x_{ij}\right) \tag{1}$$

$$\text{s.t.} \quad \sum_{j=1}^{N_t} x_{ij} \leq L_t \quad \forall i \in \mathcal{I} \tag{2}$$

$$\sum_{i=1}^{N_u} x_{ij} \leq 1 \quad \forall j \in \mathcal{J} \tag{3}$$

$$\sum_{i=1}^{N_u}\sum_{j=1}^{N_i} x_{ij} = N_{\min} \triangleq \min\left\{N_t, N_u L_t\right\} \tag{4}$$

$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in \mathcal{I} \times \mathcal{J} \tag{5}$$

Here, decision variable $\mathbf{x}_{ij} = 1$ if task j is assigned to agent i, and is set to 0 otherwise. $\mathbf{x}_i \in \{0,1\}^{N_t}$ is then a vector with $\mathbf{x}_{ij}$ as jth element. Index sets are defined as $\mathcal{I} \triangleq \{1,\dots,N_u\}$ for the agents, and $\mathcal{J} \triangleq \{1,\dots,N_t\}$ for the tasks. Vector $\mathbf{p}_i \in (\mathcal{J} \cup \{\emptyset\})^{L_t}$ then represents an ordered sequence of tasks for agent i; its kth element is $j \in \mathcal{J}$ if agent i conducts j at the kth point along the path, and becomes $\emptyset$ (denoting an empty task) if agent i conducts less than k tasks. The local reward for agent i is therefore represented by the summation term in Equation (1). It should be noted that in this formalization, the allocation operates around a reward function, resulting in a maximization problem. This score function is usually assumed to satisfy $c_{ij}(\mathbf{x}_i, \mathbf{p}_i) \geq 0$ and can be any (usually non-negative) function of either assignment $\mathbf{x}_i$ and/or path $\mathbf{p}_i$. In the case of mobile autonomous vehicles and robots, scoring/cost functions often exploit path-dependent properties to represent the cost/reward of taking on various tasks (e.g. path length, mission completion time) [17].

3.2 Consensus-based MRTA Algorithms

Consensus-based methods, a subset of market-based methods, leverage peer-to-peer exchanges of information combined with auction logic to determine an allocation in a decentralized and distributed fashion [3]. Contrary to conventional auctions, such methods decentralize the winner determination problem, and thereby avoid a central single point of failure, as illustrated in Figure 2.

CBBA, the most representative algorithm of this category, determines the task allocation through each agent locally determining bids, sharing them with their neighbors, and determining the assignment locally from all the information received. Such algorithms therefore alternate between two phases; an auction phase (during which bids are computed locally), and a consensus phase (during which the results of the auction phase, referred to as the winning bids lists, are shared and conflict resolution is performed to converge to a global allocation of tasks). In CBBA, tasks are evaluated using their respective added value in a local plan constructed by each agent.

3.2.1 *Phase 1: Auction.*

In the CBBA algorithm auction phase, agents construct bundles of tasks $\mathbf{b}_i$. Each agent continuously adds tasks to its bundle $\mathbf{b}_i$ until all tasks it is capable of including have been integrated. Tasks are added based on their marginal score improvement c_{ij}, which is determined by comparing the reward value of performing the tasks at specific locations in the current path $\mathbf{p}_i$. This path is therefore an ordered list of tasks in the agent's bundle $\mathbf{b}_i$, inserted sequentially into $\mathbf{p}_i$ at the location that maximizes the inserted task's marginal gain. The bundle and path are recursively updated until the maximum assignment size is reached or no further tasks are available to be added to the bundle. Upon having constructed a bundle $\mathbf{b}_i$/path $\mathbf{p}_i$ pair, the agents share the marginal gains obtained from inserting each task in the bundle/path with the other agents, which then release tasks which contribute a smaller marginal gain to their bundle. Note that when a task is released, the agent must then drop all tasks that were added after it in their respective bundle (all the tasks following the dropped task in bundle $\mathbf{b}_i$). The agents then complete their bundle by sequentially re-computing the new marginal gains obtained from including additional tasks in their bundle. This process is repeated until all tasks have been allocated to the agents that achieve the largest marginal gain when including them in their respective bundles.

Each agent therefore maintains five vectors: a winning bid list $\mathbf{y}_i$ (of size N_t, corresponding to the largest marginal gain observed across the fleet for each task), a winning agent list $\mathbf{z}_i$ (of size N_t, tracking the agent corresponding to the marginal gains listed in $\mathbf{y}_i$), a timestamp list $\mathbf{s}_i$ (of size N_u, a list of timestamps recording the last contact with each agent in the fleet used for consensus conflict resolution), a bundle $\mathbf{b}_i$ and corresponding path $\mathbf{p}_i$.

The complete pseudo-code describing this process can be seen in [3, Algorithm 3].

3.2.2 *Phase 2: Consensus.*

The purpose of this phase is to converge to a consensus on a single list of winning bids across the agent collective, which is in turn used to determine the winners and subsequent task allocation. CBBA defines $\mathbb{G}(\tau)$ as the undirected communication network at time τ, represented by the symmetric adjacency matrix $G(\tau)$. This matrix models the presence of links between agents, such that $g_{ij}(\tau) = 1$ indicates a link between agents i and j, and 0 otherwise. Agents i and j are considered neighbors if there is a link between them. Moreover, each agent is self-connected ($g_{ii}(\tau) = 1$ for all i) by convention. At each time step τ, the consensus phase performed by each agent is decomposed into the following steps:

Step 1— Share and receive local states with neighbors: Each agent i sends its local winning bids list $\mathbf{y_i}$, list of winning agents z_i, and list last contact timestamp s_i to its neighbors and receives the equivalent from each of its neighbors.

Step 2— Update local states using ones received: The consensus process is performed for each received bid list $\mathbf{y_k}$ for all k for which $g_{ik}(\tau) = 1$. Agent i updates y_{ij} values using the values obtained from all its neighbors and the rules depicted in [3, Table 1] of the original CBBA paper.

Step 3— Lose assignment if outbid by neighbors: Each agent loses all the tasks (and subsequent tasks appearing in bundle b_i) if it finds itself outbid by another agent for the tasks currently in its bundle.

3.3 Related Works

When putting CBBA and similar approaches in the context of our *driving scenario*, it can be observed that the operator has no control over the allocation process online. It is therefore unable to assist or contribute to the process, and must entirely take back control of the robots to leverage the additional information it has at its disposal. We therefore propose a solution seeking to do so while also allowing for modulating the level of control the operator exerts on the allocation process. Our direction is close to the *shared autonomy* notion which is defined as "the autonomous control of the majority of degrees of freedom in a system while designing a control interface for human operators to control a reduced number of parameters defining the global behavior of the system" [13]. This may be done to enable controlling robots through packaging complex action sequences into abstract sequences [16], or controlling a large number of robots with few operators [18]. Additionally, a lot of research may be found on autonomous allocation of tasks for a team made up of both humans and robots [9, 19], and human monitoring of mission execution [4].

Human-robot interaction may furthermore be broken down into two key paradigms [12]: *complementary interaction* (where the human and the robots control different subsets of tasks, and the robots plan "around" the human instructions), and *overlapping interaction* (where both the human and robots control the same set of tasks). While a number of approaches exist for the former [12], little work has been done on the latter, which our work seeks to address. We therefore propose a solution for the challenge of combining human and autonomous control in a task allocation problem through the introduction of a novel mechanism in decentralised auction-based methods: bid intercession. Our approach seeks to retain underlying convergence and robustness guarantees while enabling the injection of additional expertise in the allocation process to optimise desired metrics.

4 CBBA WITH INTERCESSION

We propose a novel mechanism, called *bid intercession*, to enable control over consensus-based algorithms during execution. This section is broken down into four sub-sections: Section 4.1 provides an overview of the mechanism and illustrates it with a few elementary examples highlighting its application to our *driving scenario*. Section 4.2 and Section 4.3 detail respectively the functioning of I-CBBA's auction phase and consensus phase.

4.1 Mechanism Explanation and Illustration

Bid intercession is the process of emitting a bid *on behalf of another agent* for a given task, with the goal of increasing or reducing its weight (and consequently odds of winning) in an auction framework. In our case, an operator will emit bids on behalf of agents, who may have already placed bids based on local information and analysis. This will result in the local bids bring overridden by the operator's. This is possible given all agents in the fleet have a clear and consistent way of determining which bid to account for when conflicting bids are present for a specific agent/task combination.

This is achieved here through the introduction of a priority hierarchy. Each agent bids with a specific priority level, and the bid with the largest priority level is always adopted over other conflicting bids for a given agent and task. An additional information vector must therefore be introduced, P_ρ, defining a fixed hierarchy between agents in N_u. P_ρ is therefore a vector of length N_u containing integers representing the priority level associated with each agent i. In turn, all bids emitted will now include the priority level of the bid emitter, and additional logic (described in the later sections) is introduced to account for the bids with the largest priority level in the event of conflicts. This approach allows for influencing auction outcomes *without altering the fundamental allocation protocol.* It preserves the algorithm's complete freedom and control over the allocation process while allowing for steering bids in a specific direction to achieve a desired result. Although any agent may perform this, it will be leveraged here specifically for human control and intervention in the coordination process. *Example 1* and *2* are provided next to showcase the application of the mechanism to address the *driving scenario* defined earlier in Section 2.

Example 2 (Single Agent Intercession). *The operator wishes to reallocate the task to a specific drone , B, which is lighter and safer for this situation to prevent a potential collision. To achieve this, the operator overrides B's original bid of $b_B = 6$ with a higher bid of $b_{B'} = 11$, surpassing A's original bid of $b_A = 10$ in the auction. This new bid, $b_{B'}$, takes precedence over b_B because it is placed by the operator with a higher priority level $p_O = 1$, compared to B's original priority level of $p_B = 0$. Initially shared with any robot in the fleet, the bid is then automatically propagated to other members via the CBBA mechanism. The higher priority level p_O ensures that the interceded bid replaces B's local bid, and the CBBA mechanism reassigns the task, ultimately allocating it to B.*

Example 3 (Agent Subset Intercession). *The operator aims to ensure that a* specific type of robot*—in this case, the drones (B and C)—completes the task. This intervention enables the drones to retain flexibility in assigning task T among themselves. The operator monitors the bids generated by the fleet and intercedes by either magnifying the drones' bids (by a factor of 10, in this example) or scaling down those of ground robots (though not applied here), assigning to the interceded bids a priority of $p_O = 1$. In this scenario, the operator overrides B's bid of $b_B = 6$ with $b_{B'} = 60$ and C's bid of $b_C = 8$ with $b_{C'} = 80$ (both possessing a bid priority of $p_{A/B} = 0$). The winner of the auction is determined using the CBBA mechanism, with C securing the task since $b_{C'} = 80$ exceeds both $b_{B'} = 60$ and A's $b_A = 10$. This process is repeated each time the robots update their local bids, provided they remain in contact with the operator. If contact is lost, the last interceded bid remains active, as the robots lack the priority levels to override it. This situation could be further refined with mechanisms such as weighted or conditional intercession rules (not covered in this paper).*

The behavior described in the *group intercession example* is used as main test case study in the next sections.

Algorithm 1: I-CBBA Auction Phase

1 **Procedure** *BUILD BUNDLE*
2 $(\mathrm{y}_i(t-1), \mathbf{z}_i(t-1), \mathrm{b}_i(t-1), \mathrm{p}_i(t-1), \mathbf{f}_i(t-1), \phi_i(t-1)$
3 $\mathrm{y}_i(t) = \mathrm{y}_i(t-1); \mathbf{z}_i(t) = \mathbf{z}_i(t-1)$
4 $\mathbf{b}_i(t) = \mathrm{b}_i(t-1); \mathbf{p}_i(t) = \mathrm{p}_i(t-1)$
5 $\mathbf{f}_i(t) = \mathbf{f}_i(t-1); \phi_i(t) = \phi_i(t-1)$;
6 **while** $|\mathbf{b}_i| < L_t$ **do** `// while there are tasks not in bundle`
7 $c_{ij} = \max_{n \leq |\mathrm{p}_i|} S_i^{\mathrm{p}_i \oplus_n \{j\}} - S_i^{\mathrm{p}_i}, \forall j \in \mathcal{J} \backslash \mathbf{b}_i$ `// compute largest marginal scores`
8 $\mathrm{merge}(f_{iji}(t), c_{ij}, \phi_{iji}(t), N_\rho(i)), \forall j \in \mathcal{J} \backslash \mathbf{b}_i$ `// merge computed marginal scores into f`
9 $h_{ij} = \mathbb{I}\left(f_{iji}(t) > y_{ij}\right), \forall j \in \mathcal{J} \backslash \mathbf{b}_i$ `// determine v.tasks`
10 **if** $|\mathbf{h}_i| = 0$ **then** `// if there are no v.tasks`
11 terminate process
12 $J_i = \mathrm{argmax}_j\, h_{ij} \cdot f_{iji}(t), \forall j \in \mathcal{J} \backslash \mathbf{b}_i$ `// get v.task with largest local bid`
13 $n_{i,J_i} = \mathrm{argmax}_n\, S_i^{\mathrm{p}_i \oplus_n \{J_i\}}$ `// determine insertion loc.`
14 $\mathbf{b}_i \leftarrow \mathrm{b}_i \oplus_{\mathrm{end}} \{J_i\}$ `// update bundle`
15 $\mathrm{p}_i \leftarrow \mathrm{p}_i \oplus_{n_{i,J_i}} \{J_i\}$ `// update path`
16 $\mathrm{y}_{i,J_i}(t) \leftarrow f_{iJ_i i}$ `// update winning bids`
17 $\mathrm{z}_{i,J_i}(t) \leftarrow i$ `// update winning agents`

Algorithm 2: Merging two values (v and w) depending on their priorities (p and q)

1 **Procedure** merge(v, w, p, q)
2 $v \leftarrow \begin{cases} w & \text{if } q > p, w \neq 0 \\ v & \text{if } q < p \\ \text{apply tie-breaker} & \text{if } q = p \end{cases}$
3 $p \leftarrow \begin{cases} q & \text{if } q > p, w \neq 0 \\ p & \text{if } q < p \\ \text{apply tie-breaker} & \text{if } q = p \end{cases}$

4.2 Phase 1: Auction Process

In order to enable bid intercession, the algorithm described in Section 3.2 needs to be extended as follows.

The base vectors $\mathbf{y}_i$, $\mathbf{z}_i$, $\mathbf{s}_i$, $\mathbf{b}_i$, and $\mathbf{p}_i$ remain the same, and two new matrices are introduced. The *current fleet bids matrix* $\mathbf{f}_i$ of size $N_t \times N_u$ is used to store the most up-to-date estimation of the *highest priority/value bids made across the fleet for each task and each agent.* In other words, f_{ijr} is the highest priority/value bid known by agent i to have been placed for task j for agent r across the fleet. The ϕ_i matrix, also of size $N_t \times N_u$, corresponds to the priority level associated with each bid figuring in $\mathbf{f}_i$. It is used in priority merging processes to decide which values to keep when updating the $\mathbf{b}_i$ matrix. Note that similarly to the original CBBA algorithm, it is assumed that all ties (such as when the priority levels are not sufficient to determine the winner ($P_\rho(i) = \beta_{ijr}$)) are resolved systematically. The auction phase can be seen in Algorithm 1, and operates as follows:

While the length of bundle $\mathbf{b}_i$ is less than the number of tasks currently available, do: **Step 1 [l7] — Computes the largest marginal score c_{ij} achieved by inserting each remaining task j not present in bundle b_i in the path p_i position yielding the largest marginal gain:**

This step remains unchanged compared to the base CBBA algorithm. A vector $\mathbf{c}_i$ representing the marginal gain contributed from inserting each task j individually in the existing path $\mathbf{p}_i$ at a position maximizing the gain magnitude is computed.

Step 2 [l8] — The computed $\mathbf{c}_i$ vector is priority merged (Algorithm 2) into the f_i matrix using the agent's priority level $N_\rho(i)$ with the priority matrix ϕ_i associated with the f_i matrix.

Step 3 [l9] — The current bids matrix f_i is compared to the winning bids y_i to generate the list of valid tasks h_i: The valid tasks $\mathbf{h}_i$ are generated using $h_{ij} = \mathbb{I}\left(f_{iji} > y_{ij}\right), \forall j \in \mathcal{J}$. It is important to note here that f_{iji} is considered and not c_i. This is necessary to ensure that the bid intercessions performed by other agents with larger priorities (than that of agent i) are the ones considered in the auction process.

Step 4 [l10-11]— The process is *terminated* if no valid tasks remain to be added to the bundle: If the agent is unable to outbid any other agents ($|\mathbf{h}_i| = 0$), either by being unable to produce larger marginal gains (if no intercession with a larger priority is present) or the current intercession value forces the agent to be outbid, the process is terminated.

Step 5 [l12-17] — The valid task with the highest bid f_{iji} is selected. It is inserted in the path at the location n_{i,J_i} yielding the largest marginal gain and the winning bids are updated. The task is added at the end of the bundle $\mathbf{b}_i$ and inserted at the most optimal location in the path $\mathbf{p}_i$, and the winning bids list $\mathbf{y}_i$ and winning agent list $\mathbf{z}_i$ are updated the same way as it is done in the base CBBA algorithm.

4.3 Phase 2: Consensus Process

Two modifications are necessary here to adapt the original consensus phase to support bid intercession. First, f_i and ϕ_i matrices are shared alongside the y_i, z_i and s_i. The updating process must then be extended to first merge the received f_k and ϕ_k matrices with local f_i and ϕ_i matrices following the prioritization logic (mentioned previously in the auction phase section). This must be performed *before* applying the consensus rules depicted in [3, Table 1]. Additionally, unlike in the base merge function, an agent also drops task j and all the subsequent ones appearing in its bundle b_i from both its bundle b_i and path p_i if its local current bids matrix f_i is updated for a task j:

$$b_i, p_i \leftarrow b_{i,:j-1}, p_i \ \ \backslash \ \ b_{i,j:} \text{ if } q > p \text{ and } q \neq 0 \tag{6}$$

This is necessary to ensure that in the event of an intercession with a smaller bid, the agent correctly releases the task (and consequently all the ones that follow since their marginal gains were calculated on the basis of the presence of task j in the bundle). This procedure is referred to as merge_r, which consists in merge with equation (6) before line 2.

Finally, and similarly to the base CBBA paper and the prioritisation logic, it is assumed that all ties occurring in the determination of either J_i in the auction phase, or z_{i,J_i} in the consensus phase are resolved systematically.

4.4 Convergence

The algorithm's convergence is unaffected by the introduction of bid intercession, as its nature allows for it to inherit the underlying decentralized method's existing guarantees [3, section V-D]. Bid intercession simply adjusts the agents' scoring process without altering the core consensus mechanism, ensuring consistent convergence across all agents.

5 EXPERIMENTAL EVALUATION

To evaluate the validity and applicability of bid intercession, a few synthetic test cases are considered. We seek through this representation to evaluate scenarios where one human operator with higher context awareness seeks to steer the allocation process in a specific direction to ensure a more efficient outcome. The dynamic implemented here is the one described in Example 3.

5.1 Experimental Setup

We consider 4 robots and 50 tasks, positioned in a 20x20 grid, from which we removed 10% of the edges (Figure 1). The goto tasks may be performed by any agent, and completing them can lead to discovering a new action task. Those are either a_1 tasks (observations), which may only be performed by agents equipped with skill s_1, or a_2 tasks (resupply), which equivalently require the skill s_2. It is possible that no action task is found upon completing a goto, which is denoted by $a_\emptyset$. *Agents* are then created with different skill sets (common goto skill, and one from $\{a_1, a_2\}$), resulting in two types of agents, and .

The *robot agents* possess a priority of 0, and evaluate their own bids as the inverse of the shortest path length between their current locations and a given task's location. Note that if any agent is not equipped with the required skill for performing a task, its bid defaults to 0 (it is not possible to win a task with a bid of 0, this ensures only agents capable of performing a task are considered in the auction). The *interceding agent* possesses a priority of 1, and unlike robots, it is capable of anticipating some action tasks (if any) at a given goto task destination. For a given task, the interceding agent will bid for all robots capable of performing a task and its subsequent action task (if any), and produce bids magnitude larger than the largest possible ones computed by all other agents (1/40 for a 20×20 grid). This agent represents a human operator injecting higher situation awareness into the allocation process. During each run, all robots start at the same location (bottom left of the grid), and after an initial announcement of 5 tasks, the 45 others are announced gradually throughout the run, from closest to farthest from the starting point (note that agents have a vision radius of 1, and will detect/discover all tasks falling within it). Upon having reached a consensus on an allocation, robots move to the corresponding tasks' locations (taking the shortest path), where they possibly discover an action task. The simulation runs in epochs, with tasks having a specific release epoch, and agents being able to perform one action per epoch (move one step towards a goto task destination, or perform an action).

We consider scenario configurations with different task requirements and fleet compositions: task requirements are defined by the number $\overline{a_\emptyset}$ of $a_\emptyset$ tasks, the number $\overline{a_1}$ (resp. $\overline{a_2}$) of a_1 (resp. a_2) tasks; fleet compositions are defined by the number $\overline{s_1}$ of robots equipped with s_1 and the number of $\overline{s_2}$ robots equipped with s_2.

We evaluate I-CBBA, our implementation of I-CBBA, where intercession is applied to bids for tasks a robot can perform, and for goto tasks followed by an action task a robot can perform. Here the human operator predicts (or just acquired extra knowledge to know) which action tasks will appear after goto tasks, and thus intercedes for compatible robots. We evaluate I-CBBA at different rates of *interventionism*. We define interventionism as the rate of intervention of the operator in providing additional information through intercession. The scenarios are tested at intervals of 10%, from 0 (no intervention, equivalent to pure CBBA) all the way to 100%, where the operator intervenes for every single task encountered.

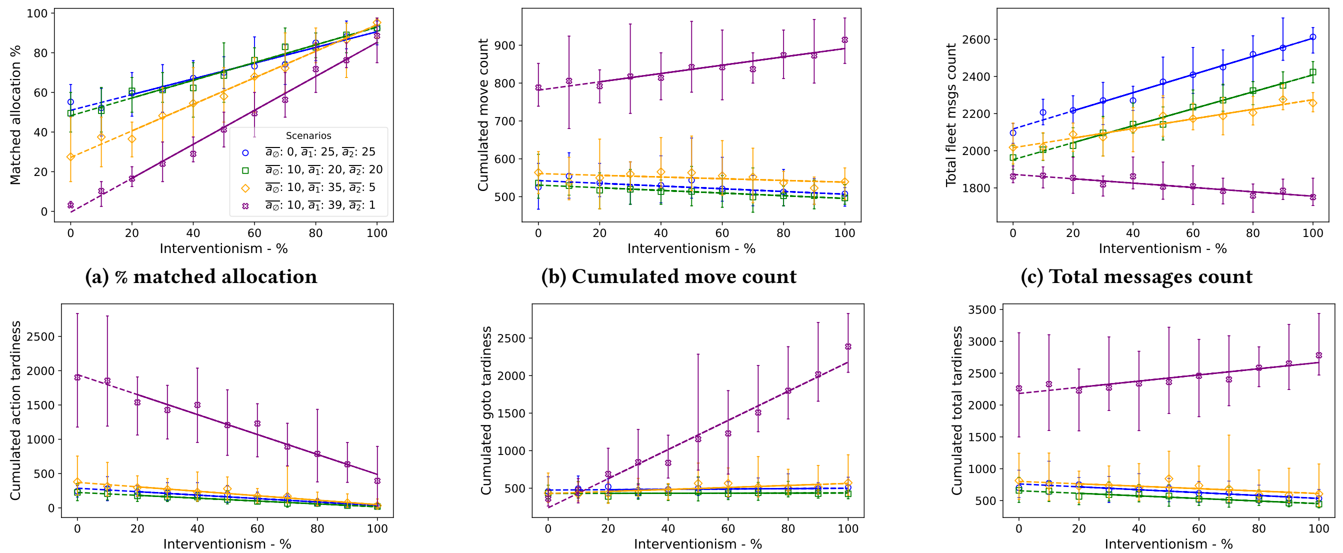
Finally, five key metrics are defined: *total step count* is the fleet's overall distance travelled, *total tardiness* measures the elapsed time between task discovery and completion, and *total goto tardiness* and *total action tardiness* track the accumulated delays for respective task types. Lastly, *total message count* is the cumulative number of messages exchanged during operations. The *% match allocation* (ratio of instances where an agent undertook a goto task and its subsequent action task) is also tracked to better observe and understand the influence of bid intercession on the allocation process.

5.2 Results Analysis

The results from running simulated experiments using ROS 2 [10] across 10 different task schedules for all configurations (totaling 880 individual runs) are presented in Figure 3. Each data point in the plots represents the mean across all runs for a given scenario, with corresponding min/max observed for given metric/interventionism combinations. The following observations can be made.

Focusing on Figure 3a, a direct linear correlation can be observed between the interventionism rate and the % matched allocation. This metric is at its lowest at the 0% interventionism mark, with the more unbalanced scenarios scoring the lowest (scenario $(10, 39, 1)$ scoring as little as 3% matched allocation initially). All scenario results significantly improve with interventionism, with the lowest mean reached being 76.25% matched allocation for $(10, 39, 1)$, and the largest being 89% for the $(10, 39, 1)$. This relation can be attributed to the fact that each operator intervention contributes additional information specifically aimed at improving this metric.

Focusing on Figure 3d, a clear negative correlation can be observed between interventionism and cumulated action tardiness. This is easily explained by the fact that the additional information provided through operator intervention allows for more effective task allocation. It ensures that an agent only completes a goto task if it is capable of completing the action task that follows. This results in a drop in action tardiness to as little as 34 at 100% interventionism (representing a drop of 84.9% from 0% interventionism). Non-zero action tardiness may be attributed to two factors: opportunity completions (when a goto task is on the way), or a shortage of one specific type of action task (in such scenario, and although



(a) % matched allocation **(b) Cumulated move count** **(c) Total messages count**

(d) Cumulated action tardiness **(e) Cumulated goto tardiness** **(f) Cumulated total tardiness**

Figure 3: Mean over 10 runs of our reference metrics vs. interventionism, for each of the four considered scenario configurations using I-CBBA. Note that an interventionism rate of 0 is equivalent to using pure CBBA.

unlikely, an incorrectly skilled agent may get assigned a task it is not specialized for as it does not have other tasks to prioritize). With the exception of the most unbalanced scenario, the % matched allocation achieved was lowest at 100% interventionism and does not fluctuate majorly as interventionism increases. These results demonstrate that the underlying consensus process is able to effectively leverage this additional information in the allocation process and enables the operator to target a specific aspect of the allocation if desired, without refactoring the whole consensus process.

Intercession encourages the construction of bundles prioritizing minimization of action tardiness over distance when possible. Consequently, the improvements in action tardiness seen in Figure 3d are a direct result of the trade-off of higher goto tardiness, resulting in the positive linear correlations observed in some of the scenarios in Figure 3e. This is a consequence of agents favoring the closest goto tasks which are followed by action tasks they are capable of immediately completing, as opposed to solely the closest tasks. Runs with 0% interventionism (equivalent to pure CBBA) therefore often achieved the lowest goto tardiness. It is also noted that this gap grows as scenarios become more unbalanced, with the $(10, 39, 1)$ scenario resulting in an increase of 460% in goto tardiness.

Looking at Figure 3f, the results show that in such trade-offs, the reduction in action tardiness generally effectively outweighs the resulting increase in goto tardiness (with the reductions in total tardiness going from 23.6% for the $(0, 25, 25)$ scenario to as much as 30% for the $(10, 20, 20)$ scenario), with the largest interventionism rates consistently minimizing the total tardiness across all instances but the last extreme case (where pure CBBA generated a total tardiness 17.2% smaller on average). This may be attributed to the under-utilization of available resources, and consequently a significant degradation of goto tardiness occurs when solely a single agent is capable of taking on the majority of tasks –limiting the fleet to operate as if it was composed of a single agent. Note that re-balancing the ratio of action tardiness to goto tardiness through limiting bundle sizes could enable unskilled agents to complete goto tasks, but would not improve total tardiness, as savings in goto tardiness would directly transfer to action tardiness.

Despite shifting optimization goals, the results seen in Figure 3b suggest that the algorithm is still able to optimize the base metric of distance. The difference in performance here is limited, with most case study variants falling within the margin of error from each other. It is important to note however that while the total move count is maintained and often improved, it is achieved while also optimizing for tardiness (e.g. the results of the full intercession configuration for scenario $(10, 35, 5)$ suggesting a reduction of 7.3% in step count in addition to the reduction of 80.6% in action tardiness). This is a direct result of the choice of bid method more than the process of intercession itself, but it nevertheless demonstrates that intercession effectively provides the ability to adjust the evaluation when worthwhile while maintaining overall performance to achieve a general improvement over specific target metrics.

These improvements however come at the cost of additional communication, as seen in Figure 3c. Intercession increases message count, with 0% interventionism (pure CBBA) consistently resulting in the lowest message exchanges. Higher intercession generally linearly relates to an increase in message count.

Figures 4 and 5 represent two instances timelines for the scenario $(10, 20, 20)$. The top half of each plots graphs the various tardiness and their evolution across the run, and the bottom visualize the tasks backlog at each time step. The figure's analysis demonstrate that they effectively support the observations made in Figure 3.

Focusing on the aggregated tardiness per epoch plots, the effect of intercession can be directly observed, with the action tardiness values being almost completely flattened out when interceded on.

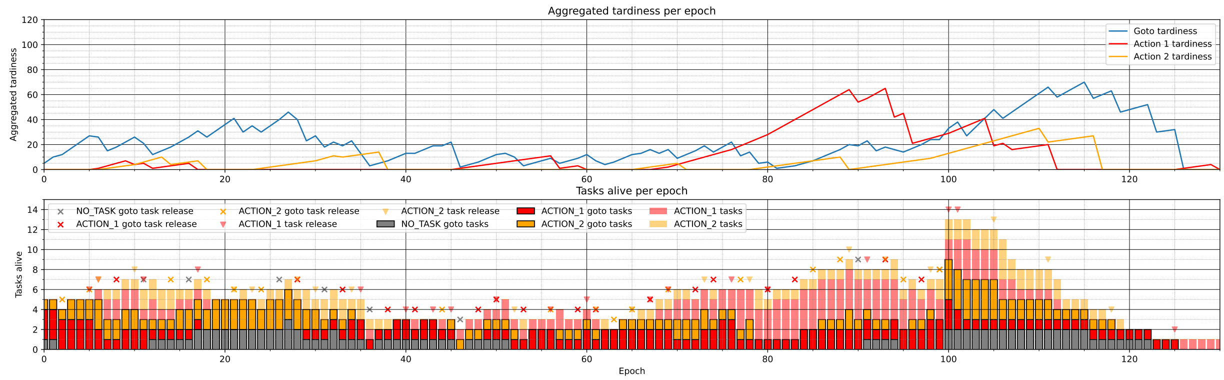


Figure 4: Instance of scenario ($\overline{a_\emptyset}$ = 10, $\overline{a_1}$ = 20, $\overline{a_2}$ = 20), with $\overline{s_1}$ = 2, $\overline{s_2}$ = 2, 0% interventionism (base CBBA).

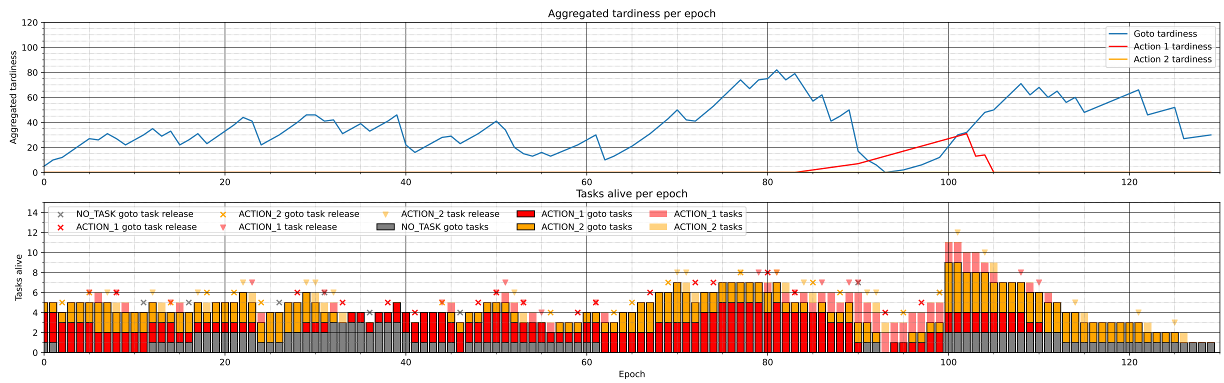


Figure 5: Instance of scenarios ($\overline{a_\emptyset}$ = 10, $\overline{a_1}$ = 20, $\overline{a_2}$ = 20), with $\overline{s_1}$ = 2, $\overline{s_2}$ = 2, 100% interventionism.

This can be well observed by looking at the action tardiness curves, with both a_1 and a_2 tardiness being flattened out in the 100% interventionism configuration.

The increase in action tardiness despite intercession (observed at epoch 84 in Figure 5) can be attributed to two key factors: the goto tasks location separation increase due to the circular release of tasks and the un-even task load ratio of type a_1 and a_2. The uneven ratio means that agents with no tasks matching their skillsets are more likely to take on other tasks (this specific scenario presenting a denser schedule of tasks a_1 around the middle of the run, whereas the a_2 tasks are more clustered around the beginning, where the distances are smaller), and the larger task spread further increases the possibility of an agent incorrectly completing a goto task. Furthermore increasing in distance means that agents require more timesteps to move to the target locations, directly translating into larger consequences for an incorrect goto completion.

6 CONCLUSIONS

We have introduced a new mechanism, *intercession*, to enable overlapping control in decentralized auction-based methods while preserving the robustness and convergence guarantees of the original method. Given a simplified MRTA problem of goto and subsequent action tasks for a fleet of agents (all heterogeneous in nature), the findings demonstrate the injection of enhanced situational awareness by the operator, resulting in a consistent reduction of at least 66.6% in total action tardiness across all tests, albeit with increased communication load. Those results were achieved with minimal degradation to our goto tardiness in the exception of some specific extreme edge cases, and while retaining similar performance in total distance traveled compared to original CBBA.

These results lead us to conclude that bid intercession is a viable and compelling method for effectively enabling agents to integrate and leverage additional information provided by the interceding agent (our operator) to enhance their autonomous allocation process while retaining the robustness guarantees of the underlying mechanism. This opening in mixed-initiative presents new collaboration possibilities between humans and autonomous teams, and provides a powerful tool for constructing complex yet seamless coordination architectures effectively leveraging the strengths of the respective actors. Understanding well these possibilities and the associated limitations may prove crucial to best take advantage of such mechanisms. Additionally, testing the performance, dynamics, and resilience of such mechanisms on additional scenarios, such as ones tailored to investigate communication failure or different intercession architectures would help further enhance our understanding and understand when, where, and how to best take advantage of such methods. The initial findings presented here, generated using ROS2, lay the groundwork for future on-ground experiments and contribute to the advancement of MRTA for operational applications.

ACKNOWLEDGMENTS

This work has been funded by the French Ministry of Defence's Innovation Agency (AID).

REFERENCES

[1] J.E. Allen, C.I. Guinn, and E. Horvtz. 1999. Mixed-initiative interaction. *IEEE Intelligent Systems and their Applications* 14, 5 (1999), 14–23. https://doi.org/10.1109/5254.796083

[2] Manolis Chiou, Nick Hawes, and Rustam Stolkin. 2021. Mixed-initiative Variable Autonomy for Remotely Operated Mobile Robots. *J. Hum.-Robot Interact.* 10, 4, Article 37 (Sept. 2021), 34 pages. https://doi.org/10.1145/3472206

[3] Han-Lim Choi, Luc Brunet, and Jonathan P. How. 2009. Consensus-Based Decentralized Auctions for Robust Task Allocation. *IEEE Transactions on Robotics* 25, 4 (2009), 912–926. https://doi.org/10.1109/TRO.2009.2022423

[4] Abhinav Dahiya, Yifan Cai, Oliver Schneider, and Stephen L. Smith. 2023. On the Impact of Interruptions During Multi-Robot Supervision Tasks. In *International Conference on Robotics and Automation (ICRA)*. IEEE, London, UK, 9771–9777. https://doi.org/10.1109/ICRA48891.2023.10160323

[5] Brian P. Gerkey and Maja J. Matarić. 2004. A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems. *The International Journal of Robotics Research* 23, 9 (2004), 939–954. https://doi.org/10.1177/0278364904045564

[6] Aaron W. Johnson, Charles M. Oman, Thomas B. Sheridan, and Kevin R. Duda. 2014. Dynamic task allocation in operational systems: Issues, gaps, and recommendations. In *IEEE Aerospace Conference*. IEEE, Big Sky, MT, USA, 1–15. https://doi.org/10.1109/aero.2014.6836205

[7] Sven Koenig, Craig A. Tovey, Michail G. Lagoudakis, Evangelos Markakis, David Kempe, Pinar Keskinocak, Anton J. Kleywegt, Adam Meyerson, and Sonal Jain. 2006. The power of sequential single-item auctions for agent coordination. In *AAAI Conference on Artificial Intelligence (AAAI)*. AAAI Press, Boston, MA, USA, 1625–1629.

[8] G. Ayorkor Korsah, Anthony Stentz, and M. Bernardine Dias. 2013. A comprehensive taxonomy for multi-robot task allocation. *Int. J. Rob. Res.* 32, 12 (Oct. 2013), 1495–1512. https://doi.org/10.1177/0278364913496484

[9] Martina Lippi, Paolo Di Lillo, and Alessandro Marino. 2023. A Task Allocation Framework for Human Multi-Robot Collaborative Settings. In *International Conference on Robotics and Automation (ICRA)*. IEEE, London, UK, 7614–7620. https://doi.org/10.1109/ICRA48891.2023.10161458

[10] Steven Macenski, Tully Foote, Brian Gerkey, Chris Lalancette, and William Woodall. 2022. Robot Operating System 2: Design, architecture, and uses in the wild. *Science Robotics* 7, 66 (2022), eabm6074. https://doi.org/10.1126/scirobotics.abm6074 arXiv:https://www.science.org/doi/pdf/10.1126/scirobotics.abm6074

[11] Robin R. Murphy, Satoshi Tadokoro, Daniele Nardi, Adam Jacoff, Paolo Fiorini, Howie Choset, and Aydan M. Erkmen. 2008. Search and Rescue Robotics. In *Springer Handbook of Robotics*, Bruno Siciliano and Oussama Khatib (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 1151–1173. https://doi.org/10.1007/978-3-540-30301-5_51

[12] Selma Musić and Sandra Hirche. 2017. Control sharing in human-robot team interaction. *Annual Reviews in Control* 44 (2017), 342–354. https://doi.org/10.1016/j.arcontrol.2017.09.017

[13] Jorge Peña Queralta, Jussi Taipalmaa, Bilge Can Pullinen, Victor Kathan Sarker, Tuan Nguyen Gia, Hannu Tenhunen, Moncef Gabbouj, Jenni Raitoharju, and Tomi Westerlund. 2020. Collaborative Multi-Robot Search and Rescue: Planning, Coordination, Perception, and Active Vision. *IEEE Access* 8 (2020), 191617–191643. https://doi.org/10.1109/ACCESS.2020.3030190

[14] Félix Quinton, Christophe Grand, and Charles Lesire. 2023. Market Approaches to the Multi-Robot Task Allocation Problem: a Survey. *Journal of Intelligent and Robotic Systems* 107, 2 (2023), 31. https://doi.org/10.1007/s10846-022-01803-0

[15] Sarvapali D. Ramchurn, Trung Dong Huynh, Feng Wu, Yuki Ikuno, Jack Flann, Luc Moreau, Joel E. Fischer, Wenchao Jiang, Tom Rodden, Edwin Simpson, Steven Reece, Stephen Roberts, and Nicholas R. Jennings. 2016. A disaster response system based on human-agent collectives. *Journal of Artificial Intelligence Research* 57, 1 (9 2016), 661–708.

[16] Casper Schou, Rasmus Skovgaard Andersen, Dimitrios Chrysostomou, Simon Bøgh, and Ole Madsen. 2018. Skill-based instruction of collaborative robots in industrial settings. *Robotics and Computer-Integrated Manufacturing* 53 (2018), 72–80. https://doi.org/10.1016/j.rcim.2018.03.008

[17] José Ricardo Sánchez-Ibáñez, Carlos J. Pérez-del Pulgar, and Alfonso García-Cerezo. 2021. Path Planning for Autonomous Mobile Robots: A Review. *Sensors* 21, 23 (2021), 29. https://doi.org/10.3390/s21237898

[18] Eugene M. Taranta II, Adam Seiwert, Anthony Goeckner, Khiem Nguyen, and Erin Cherry. 2023. From Warfighting Needs to Robot Actuation: A Complete Rapid Integration Swarming Solution. *Field Robotics* 3, 1 (Jan. 2023), 460–515. https://doi.org/10.55417/fr.2023015 arXiv:2304.07651 [cs].

[19] Yang You, Vincent Thomas, Francis Colas, Rachid Alami, and Olivier Buffet. 2023. Robust Robot Planning for Human-Robot Collaboration. In *International Conference on Robotics and Automation (ICRA)*. IEEE, London, UK, 9793–9799. https://doi.org/10.1109/ICRA48891.2023.10161406