

Solving Organizational Multi-Robot Task Allocation Problems with Consensus-based Auctions

Victor Guillet

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

Charles Lesire

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

Christophe Grand

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

Gauthier Picard

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

ABSTRACT

We propose a new approach to multi-robot task allocation that explicitly incorporates fleet organizational structures and constraints. We first formalize the Organizational Multi-Robot Task Allocation (Org-MRTA) problem, which extends classical MRTA by requiring allocations to respect organizational roles, missions, and norms in addition to performance. To address this problem, we introduce Org-CBBA, a hierarchical extension of the Consensus-Based Bundle Algorithm (CBBA), leveraging a MOISE+ organizational model to guide the allocation process to produce good quality solutions that remain organizationally coherent. This approach preserves the distributed robustness and convergence guarantees of CBBA while aligning task allocations with organizational doctrine. We evaluate Org-CBBA in synthetic mission scenarios inspired by real deployments and show that it maintains allocation quality while reducing computational overhead, while ensuring that solutions respect organizational requirements and constraints.

KEYWORDS

Multi-Robot Task Allocation; Organizational Modeling; Consensus-based Auctions

1 INTRODUCTION

Future space missions increasingly deploy fleets of autonomous and semi-autonomous assets, including satellite constellations and cooperative robotic teams. These missions demand rapid coordination and adaptability under severe communication and supervision constraints, rendering continuous centralized control impractical. Simultaneously, execution must comply with predefined command structures derived from planning, safety, and organizational doctrine. The organization is therefore intrinsic to mission design, and multi-agent systems must satisfy not only performance and robustness criteria but also explicit organizational constraints that delimit feasible allocations.

A central issue is the Multi-Robot Task Allocation (MRTA) problem, which assigns heterogeneous tasks to heterogeneous agents [4]. The objective is to optimize performance, such as minimizing mission duration or maximizing coverage, while accounting for capabilities, task dependencies, and environmental conditions. MRTA approaches range from centralized methods, which can provide globally optimal solutions but are unsuitable under unreliable communication, to distributed methods based on local decisions

and peer-to-peer exchanges [11, 12]. The latter are generally more robust and operationally realistic.

Among distributed approaches, consensus-based algorithms are prominent. The Consensus-Based Bundle Algorithm (CBBA) [3] alternates local bundle construction with peer-to-peer consensus, ensuring distributed execution, robustness to communication failures, and convergence to a conflict-free allocation. However, CBBA and its variants are flat, with all agents bidding on all tasks, limiting scalability, and are role-agnostic, lacking explicit organizational structure.

A natural extension is hierarchical allocation. Existing solutions often cascade distinct algorithms [1], such as team-level followed by agent-level allocation, but this decoupling weakens integration. A unified hierarchical consensus mechanism can improve scalability by restricting bidding to role-relevant tasks and align allocations with explicit organizational models, enabling reasoning at multiple abstraction levels.

To represent the organization, we adopt MOISE+ [7], which specifies: (i) a structural model of roles, groups, and hierarchies; (ii) a functional model of missions, goals, and plans; and (iii) a deontic model linking roles to missions via permissions and obligations. This provides hierarchical abstraction and normative consistency.

We introduce Org-CBBA, a hierarchical consensus-based allocation algorithm embedding MOISE+ specifications into the allocation process. Org-CBBA preserves CBBA’s distributed robustness and convergence guarantees while incorporating organizational abstraction and hierarchical workload distribution, producing scalable and organizationally coherent allocations. The paper makes three contributions: (i) the Org-MRTA model extending MRTA with organizational constraints; (ii) the Org-CBBA algorithm solving Org-MRTA; and (iii) an experimental evaluation on a scenario from a robotic challenge.

The remainder of the paper is structured as follows. Section 2 presents a motivating scenario; Section 3 reviews MRTA, consensus-based allocation, and MOISE+; Section 4 formalizes Org-MRTA; Section 5 details Org-CBBA; Section 6 reports experimental results in synthetic scenarios; and Section 7 concludes and outlines future work.

2 MOTIVATING MULTI-ROBOT SCENARIO

We consider a planetary surface exploration mission with a heterogeneous fleet of autonomous robots deployed from a central

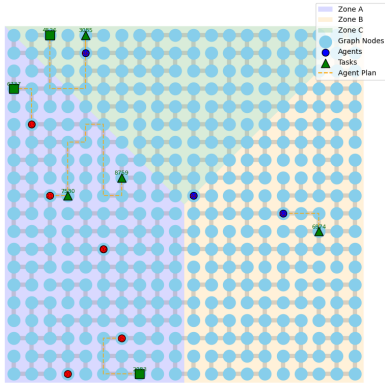


Figure 1: Environment and organizational layout.

landing site. Organizational roles and responsibilities are defined prior to deployment, while task allocation and execution occur autonomously onboard under intermittent and delayed communication with ground control.

At scenario onset, the fleet has stabilized the area around the base and operates in two geographically separated forward exploration zones extending in different directions. Transitions between zones are costly due to terrain and traversal constraints. Although no longer a primary exploration target, the base area remains operationally critical for shared logistics and support. Robots must therefore sustain exploration in both forward zones while collectively maintaining rear-area functions.

This setting induces a two-team organization: each team is primarily assigned to one forward zone to reduce costly cross-zone motion, while both share responsibility for logistics near the base. This structure captures mission planning trade-offs among spatial separation, coordination efficiency, and operational predictability under constrained communication.

From this abstract mission, we derive a simplified reference scenario that preserves the key spatial and organizational features. The objective is not to replicate a specific deployment, but to provide a controlled testbed for evaluating task allocation under explicit organizational constraints.

The environment (Figure 1) is represented as a partially connected 20×20 grid of nodes. A central landing site is located near the middle of the grid, from which all robots initially depart. The surface is divided into three exploration zones (A, B, and C), each associated with different operational assumptions: **Zone A (primary exploration zone)**: the highest-priority region, characterized by the greatest density of exploration opportunities. Tasks in this zone primarily involve surface mapping, localized inspection, and anomaly characterization. **Zone B (secondary exploration zone)**: a lower-intensity region requiring persistent monitoring and periodic exploration, including mapping and deployment of scientific instruments. **Zone C (shared exploration zone)**: a tertiary exploration region with lower task density, representing partially explored areas that remain scientifically relevant. This zone has no fixed team ownership and serves as a shared operational space.

Organizational structure. To reflect mission-level planning, the fleet is divided into two organizational groups: *Surface Team A*

and *Surface Team B*. The organizational hierarchy is strictly two-level: (1) *Surface Team A* is assigned responsibility for tasks arising in Zone A, while *Surface Team B* is assigned responsibility for tasks in Zone B; and (2) in Zone C, which represents a shared exploration region, tasks may be allocated to robots from either team. This structure enforces geographically scoped responsibilities while preserving flexibility in shared areas, providing a natural setting for evaluating how organizational constraints interact with decentralized consensus-based allocation.

Task model. Tasks appear dynamically and unpredictably throughout the mission, reflecting evolving scientific targets and environmental conditions. Their spatial distribution follows the zone priorities: the probability of task appearance is highest in Zone A, moderate in Zone B, and lowest in Zone C. Three task types are modeled, each requiring a specific capability:

- Mapping/Inspection tasks (O): reaching a location to acquire imagery or sensor data,
- Deployment tasks (T): reaching a location to deploy a scientific instrument or containment device,
- Anomaly-response tasks (I): reaching a location to inspect or approach a detected anomaly.

Although semantically distinct, all tasks are uniformly represented as go-to tasks: a robot must reach the designated node and possess the required skill. Tasks are modeled as independent, without sequential dependencies, allowing allocation decisions to be studied in isolation.

Fleet composition. The fleet is heterogeneous in skills but homogeneous in mobility. All robots move one step per time unit on the grid. The fleet is composed of: 3 robots capable only of mapping/inspection tasks (O), 2 robots capable of mapping and deployment tasks (O,T), 2 robots capable of mapping and anomaly-response tasks (O,I), 2 robots capable of deployment and anomaly-response tasks (T,I). Each robot is eligible only for tasks requiring skills it possesses, ensuring that allocation reflects heterogeneous capabilities.

During extended periods without ground contact, all task allocation decisions are performed onboard through inter-robot communication, subject to the organizational constraints defined above. This scenario therefore provides a representative testbed for studying hierarchical, decentralized task allocation for surface exploration missions under communication-limited conditions.

3 BACKGROUND AND RELATED WORK

This section introduces the Multi-Robot Task Allocation (MRTA) problem and its formal mathematical definition. We then review the main classes of approaches developed to solve MRTA, with a focus on decentralized consensus-based methods. Next, we present organizational modeling as a complementary paradigm for structuring multi-agent systems, with examples of different families. Finally, we discuss existing works that attempt to bridge MRTA and organizational models.

3.1 Multi-Robot Task Allocation Problem

The Multi-Robot Task Allocation (MRTA) problem [4] addresses the challenge of assigning a set of tasks to a team of robots in a

way that maximizes performance (e.g., mission success, reward) or minimizes cost (e.g., travel time, resource consumption). An allocation is valid if each task is assigned to at most one agent, and each agent is limited by its capacity (the maximum number of tasks it can handle).

DEFINITION 1. A MRTA problem $\langle \mathcal{T}, \mathcal{A} \rangle$ involves a set \mathcal{T} of tasks and a set \mathcal{A} of agents, and consists in assigning each task $\tau_j \in \mathcal{T}$ to an agent $a_i \in \mathcal{A}$, noted $\tau_j \mapsto a_i$, whilst maximizing some performance criteria.

Considering each agent can be assigned at most L_τ tasks, and the maximum number of assignable tasks is $N_{\min} \triangleq \min\{|\mathcal{T}|, |\mathcal{A}| \cdot L_\tau\}$, MRTA can be expressed as follows [3]:

$$\max \quad \sum_{a_i \in \mathcal{A}} \left(\sum_{\tau_j \in \mathcal{T}} c_{ij}(\mathbf{x}_i, \mathbf{p}_i) x_{ij} \right) \quad (1)$$

$$\text{s.t.} \quad \sum_{\tau_j \in \mathcal{T}} x_{ij} \leq L_\tau \quad \forall a_i \in \mathcal{A} \quad (2)$$

$$\sum_{a_i \in \mathcal{A}} x_{ij} \leq 1 \quad \forall \tau_j \in \mathcal{T} \quad (3)$$

$$\sum_{a_i \in \mathcal{A}} \sum_{\tau_j \in \mathcal{T}} x_{ij} = N_{\min} \quad (4)$$

$$x_{ij} \in \{0, 1\} \quad \forall a_i \in \mathcal{A}, \tau_j \in \mathcal{T} \quad (5)$$

Here, the decision variable $x_{ij} = 1$ if $\tau_j \mapsto a_i$, and 0 otherwise. The vector $\mathbf{p}_i \in (\{1, \dots, |\mathcal{T}|\} \cup \{\emptyset\})^{L_\tau}$ represents agent i 's ordered sequence of tasks (its path). The score function $c_{ij}(\mathbf{x}_i, \mathbf{p}_i) \geq 0$ encodes the reward for agent i to perform task j , and may depend on the agent's current task path. In mobile robotics, this function typically captures path-dependent metrics such as travel distance, energy cost, or mission completion time.

A wide variety of methods have been proposed for the MRTA problem, broadly falling into two categories. *Centralized approaches:* A single planner or auctioneer computes the allocation from global knowledge, often using combinatorial optimization (e.g., MILP, Hungarian algorithm [2]) or centralized auctions. While such methods yield high-quality or optimal solutions, they scale poorly and create a single point of failure, limiting their suitability for large-scale or communication-constrained settings. *Decentralized approaches:* Here, agents iteratively exchange bids or commitments to collectively converge to an allocation. These methods are more robust and scalable, as they avoid reliance on a central entity. Consensus-based algorithms form a key family, with the *Consensus-Based Bundle Algorithm* (CBBA) [3] being the most influential. CBBA alternates between local bundle construction and peer-to-peer consensus, and has been extensively studied for its scalability, robustness, and applicability in dynamic environments. Given their resilience to communication loss and scalability to larger teams, decentralised approaches are particularly relevant for multi-robot systems deployed in real-world, uncertain environments.

Additionally, many CBBA extensions exist for tackling different constraint types [1, 8, 9, 17]. Notably, intercession mechanisms (I-CBBA [5, 6]) provides a way for agents to bid on behalf of other agents, which we will leverage for our organizational extension.

3.2 Organizational Modeling and Allocation

As multi-robot systems scale, flat peer-to-peer coordination becomes inefficient and hard to interpret. *Organizational models* address this by structuring agent societies through roles, groups, authority, and norms, constraining autonomy in a purposeful way

to ensure global objectives. The *Gaia methodology* [18] frames systems as organizations of roles, responsibilities, and protocols, offering both macro (societal) and micro (agent) views. However, Gaia assumes a *static* structure, making it more suited to design than runtime adaptation. Frameworks such as *OperA* [10, 16] distinguish between specification (roles, objectives, dependencies) and enactment (agents adopting roles). OperA supports *open systems*, where agents may join or leave dynamically, while tools like *OperettA* assist in design. Other approaches (e.g., *OMNI* [15]) combine structural and normative dimensions. In normative and institution-based models, norms (permissions, obligations, prohibitions) regulate agent behavior. Examples include *electronic institutions* and *HarmonIA* [14], which emphasize compliance and enforcement in heterogeneous or partially conflicting systems. Finally, *MOISE+* [7] integrates structural (roles, groups), functional (missions, goals), and deontic (permissions, obligations) layers. It supports complex hierarchical task structures and runtime reorganization, enabling flexibility while maintaining normative control, as needed in our scenario.

3.3 Focus on MOISE+

A MOISE+ *Organizational Specification* (OS) is defined as a tuple: $OS = \langle SS, FS, DS \rangle$, where SS is the *structural specification*, FS the *functional specification*, and DS the *deontic specification*. The *structural specification* $SS = \langle \mathcal{R}, \mathcal{SG}, \sqsubseteq, \mathcal{L}, \mathcal{C} \rangle$ defines: a set of roles $\mathcal{R} = \{\rho_1, \dots, \rho_{|\mathcal{R}|}\}$, a set of group specifications \mathcal{SG} , an inheritance relation $\sqsubseteq \subseteq \mathcal{R} \times \mathcal{R}$ ($\rho \sqsubseteq \rho'$ means ρ' specializes ρ), a set of links $\mathcal{L} = \{\ell_i(\rho_i^s, \rho_i^d, t), i \in [1..|\mathcal{L}|]\}$ with $t \in \{\text{acq, com, aut}\}$ denotes acquaintance, communication, or authority links, and a set of compatibility constraints \mathcal{C} of the form $\rho_i \bowtie \rho_j$ indicating which roles may be jointly played. The *functional specification* $FS = \langle \mathcal{G}, \mathcal{M}, \mathcal{P}, mo, nm \rangle$ defines a set of goals \mathcal{G} , missions \mathcal{M} , and plans \mathcal{P} . Each mission $m \in \mathcal{M}$ maps to a set of goals via $mo : \mathcal{M} \rightarrow \mathbb{P}(\mathcal{G})$, and $nm : \mathcal{M} \rightarrow \mathbb{N} \times \mathbb{N}$ defines the cardinality of agents required to commit to each mission. The *deontic specification* defines the norms that link roles to missions, that is a set DS of permissions (resp. obligations) of the form $\text{per}(\rho, m, tc)$ (resp. $\text{obl}(\rho, m, tc)$), where $\text{per}(\rho, m, tc)$ grants permission for role ρ to commit to mission m under time constraint tc , and $\text{obl}(\rho, m, tc)$ imposes an obligation for role ρ to commit to mission m under time constraint tc . Inheritance ensures obligations and permissions are propagated along \sqsubseteq :

$$\text{obl}(\rho, m, tc) \Rightarrow \text{per}(\rho, m, tc) \quad (6)$$

$$\rho \sqsubseteq \rho' \wedge \text{obl}(\rho, m, tc) \Rightarrow \text{obl}(\rho', m, tc) \quad (7)$$

3.4 Bridging MRTA and Organizational Models

Although MRTA methods and organizational models emerged from distinct research traditions, they address complementary aspects of multi-robot coordination. MRTA focuses on algorithmic efficiency and conflict-free allocation of tasks, while organizational models emphasize structure, authority, and norms for scalable, interpretable cooperation. Several works have attempted to bridge these perspectives.

Early MRTA taxonomies recognized the role of coalitions and hierarchies in structuring allocation [4, 11]. Hierarchical team allocation and multi-stage pipelines have been studied [1], while

some works explicitly use organizational models to guide task allocation: for example, frameworks based on *OperA* incorporate role and organizational structure to determine who can/should perform which objectives [10]. *OperA*'s specification dimension (defining objectives, role dependencies) and enactment dimension (who plays what role) provide a basis for allocation constrained by organizational roles. Role-based methodologies (*Gaia* [18] and its extensions) provide design-time structure that can restrict possible agent-task mappings (via roles, permissions, protocols), which can be leveraged in MRTA to reduce search space, enforce constraints, or improve interpretability. Nevertheless, existing approaches often remain limited: they either rely on ad-hoc organizational scaffolding without a normative layer, or integrate task allocation only loosely into organizational formalisms. This motivates our work, which seeks to develop a tighter integration between consensus-based MRTA algorithms and formal organizational specifications, thereby bridging the gap between distributed task allocation and principled organizational modeling.

4 MODEL AND PROBLEM DEFINITION

Org-MRTA extends classical MRTA by embedding an explicit organizational model (based on MOISE+) that links roles, missions, and agents. This foundation allows us to formally define Org-MRTA as an extension of MRTA subject not only to performance criteria but also to organizational validity.

4.1 Instantiating MOISE+ for a Robot Fleet

The abstract organizational model from MOISE+ is instantiated onto a concrete fleet as follows. We first define a set of *agent classes* \mathcal{AC} , each associated with a subset of skills from the whole set of skills Σ via a mapping $\sigma : \mathcal{AC} \rightarrow \mathbb{P}(\Sigma)$. A set of *agents* $\mathcal{A} = \{a_1, \dots, a_{N_a}\}$ is then introduced (same from the MRTA model), and we note $a_i \in AC_j$, with $AC_j \in \mathcal{AC}$, the fact that agent a_i is an instance of agent class AC_j . By extension to the mapping σ , we note $\sigma(a_i)$ the set of skills of agent a_i , inherited from its agent class. Each goal $g \in \mathcal{G}$ is annotated with a set of required skills $\kappa(g) \subseteq \Sigma$, and each task $\tau_j \in \mathcal{T}$ (same from the MRTA model) is an instance of a goal $g_k \in \mathcal{G}$, and thus inherits its skill requirements: $\kappa(\tau_j) = \kappa(g_k)$. Similarly, we define a set of *group instances* $\mathcal{G} = \{\gamma_1, \dots, \gamma_{N_g}\}$, each being an instance of some group specification in \mathcal{SG} . An assignment relation $\pi : \mathcal{A} \times \mathcal{G} \rightarrow \mathbb{P}(\mathcal{R})$ specifies which agent plays which role in which group instance. A role ρ responsible for a set of goals \mathcal{G}_ρ can be validly allocated to an agent $a \in \mathcal{A}$ though π only if $\bigcup_{g \in \mathcal{G}_\rho} \kappa(g) \subseteq \sigma(a)$.

This instantiation grounds the organizational specification in the fleet, linking abstract missions and goals to concrete agents, roles, and capabilities in a single model:

$$OE = \langle OS, \mathcal{T}, \mathcal{A}, \mathcal{AC}, \mathcal{G}, \Sigma, \sigma, \pi, \kappa \rangle \quad (8)$$

These specifications should ensure correctness at three levels: (i) that the organization structure and missions are well-formed with respect to MOISE+ constraints, (ii) that the fleet provides the skills required by the goals, and (iii) that assignments of agents to roles are feasible.

4.2 Organizational Multi-Robot Task Allocation

DEFINITION 2. An Organizational Multi-Robot Task Allocation (*Org-MRTA*) problem $\langle OS, \mathcal{T}, \mathcal{A}, \mathcal{AC}, \mathcal{G}, \Sigma, \sigma, \pi, \kappa \rangle$ is a classical MRTA problem $\langle \mathcal{T}, \mathcal{A} \rangle$ (Definition 1) provided with an instantiated organizational model *OE* (see Section 4.1). It consists in assigning each task $\tau_j \in \mathcal{T}$ to an agent $a_i \in \mathcal{A}$, noted $\tau_j \mapsto a_i$, while maximizing a set of performance criteria and satisfying the additional structural, functional, and deontic requirements imposed by the organization, such that:

- (1) a task τ_j can be allocated to an agent a_i only if $\kappa(\tau_j) \subseteq \sigma(a_i)$,
- (2) a task τ_j can be allocated to an agent a_i only if $\exists \gamma \in \mathcal{G}, \exists \rho \in \pi(a_i, \gamma), \exists m \in \mathcal{M}, \exists tc$ s.t. $g \in mo(m)$ and $per(\rho, m, tc)$,
- (3) a task τ_j must be allocated to agent a_i if $\exists \gamma \in \mathcal{G}, \exists \rho \in \pi(a_i, \gamma), \exists m \in \mathcal{M}, \exists tc$ s.t. $g \in mo(m)$ and $obl(\rho, m, tc)$.

Thus, Org-MRTA can be seen as a *constrained variant* of MRTA where feasible allocations are restricted by organizational roles, missions, and authority relations. These additional constraints ensure that the resulting allocation is not only optimal with respect to performance (e.g. travel distance, energy, completion time) but also *organizationally valid*.

Let us note that a solution to Org-MRTA is also a solution to the related MRTA, but the performances might differ due to the constraining organization (e.g. a complex organization might not be as efficient as a non-organized fleet for some criteria, such as the traveled distance, but better for other ones, such as the amount of exchanged messages). Moreover the allocation of roles specified in *OE* to agents is not an output of Org-MRTA. Org-MRTA focuses on task allocation, not role allocation. Finally, the current model does not exploit the planning dimension of the functional specification, since we again focus on task allocation, thus the missions and goals.

5 ORG-CBBA: ORGANIZATIONAL CONSENSUS-BASED BUNDLE ALGORITHM

Org-CBBA extends classical CBBA by embedding an explicit *organizational specification* into the allocation process. Instead of operating in a flat agent-task space, Org-CBBA exploits the hierarchical structure defined by MOISE+, linking roles, groups, missions, and goals. As a result, task allocation unfolds as a structured chain of decisions consistent with organizational constraints. This preserves the decentralized, consensus-driven nature of CBBA while ensuring allocations remain aligned with the organization.

5.1 Rationale

Formally, Org-CBBA embeds the instantiated organizational specification $OE = \langle OS, \mathcal{A}, \mathcal{AC}, \mathcal{G}, \Sigma, \sigma, \pi, \kappa \rangle$, where $OS = \langle SS, FS, DS \rangle$ (see Section 3.3, Section 4.1). At a high level, Org-CBBA operates through a *cascading sequence of auctions* that progressively narrow down responsibility for each task along the organizational hierarchy of groups of the fleet. The process unfolds as follows:

- At the *fleet level*, abstract coordination tasks are first auctioned among the fleet's top-level groups using a I-CBAA-style mechanism.
- The winning group becomes responsible for the task and, in doing so, generates a new abstract coordination task that

is auctioned at the next level of decomposition, where its child groups compete.

- The new abstract task for the substructure is then auctioned among its child groups through a I-CBAA-style auction.
- The process continues until it reaches the lowest-level group, and thus switches to a CBBA auction among the individual agents of that group. This final stage allocates the concrete execution of the task to a specific agent.

In this way, allocation answers the question – *which entity is responsible for this task?* – at progressively finer levels of resolution: from a top-level fleet group, to nested groups, and ultimately to individual agents. At the group level, bids are placed *on behalf of the group* by agents acting as its proxies through the *intercession mechanism* introduced in I-CBAA [5]. Group-level I-CBAA bids remain *lightweight and approximate*, but their resolution increases as auctions are performed over progressively smaller subsets of agents. I-CBAA is used at the group level since these auctions aim to assign responsibility rather than to sequence tasks. Task ordering is less relevant at this stage, as groups do not directly execute tasks—the temporal reasoning required for sequencing is deferred to the agent level, where CBBA auctions determine concrete allocations and execution plans.

As responsibility cascades downward, the bids therefore become more representative of the actual capabilities of the agents involved, culminating in the lowest-level CBBA auction where concrete task allocations are determined with full accuracy.

Figure 2 illustrates the previously described process for the motivating multi-robot scenario, which is also later used in the experiment section.

5.2 Org-CBBA Specific Organizational Concepts

Org-CBBA requires defining a few additional notions and extending OE to incorporate key coordination elements necessary for its correct functioning.

In SS, a single abstract *proxy role* $\text{proxy} \in \mathcal{R}$ is introduced to enable group representation during inter-group interactions. Upon instantiation, this role can be played within any group $\gamma_k \in \mathcal{SG}$, thereby acting as the proxy for that specific group. Formally, if a_i acts as proxy in γ_k , i.e. $\text{proxy} \in \pi(a_i, \gamma_k)$. The proxy role is set to be compatible with all other roles, $\forall \rho \in \mathcal{R}, \text{proxy} \bowtie \rho$, ensuring that multiple agents can play it simultaneously.

A specific *allocation mission* $m^{\text{alloc}} \in \mathcal{M}$ dedicated to task delegation between hierarchical levels is defined in FS. This mission is composed of a new *allocation goal* $g^{\text{alloc}} \in \mathcal{G}$. We denote by $\mathcal{T}_{\text{alloc}}$ the set of allocation tasks. Each $\tau^{\text{alloc}_{jk}} \in \mathcal{T}_{\text{alloc}}$ is an instance of an allocation goal $g^{\text{alloc}} \in \mathcal{G}$. \mathcal{T} is then the set of *concrete tasks*, corresponding to the executable tasks ultimately assigned to individual agents. Note that $\mathcal{T}_{\text{alloc}} \notin \mathcal{T}$ (thus $\mathcal{T}_{\text{alloc}}$ is effectively not part of the Org-MRTA problem).

We define the parents of a group γ as the set of all supergroups of γ following the inclusion relation: $\text{parents}(\gamma) = \{\gamma' \mid \gamma \subset \gamma' \text{ or } \exists \gamma'' \text{ such that } \gamma' \in \text{parents}(\gamma'') \text{ and } \gamma \subset \gamma''\}$. If $a \in \gamma$, then $\text{parents}(a) = \text{parents}(\gamma)$. Similarly, if $\tau^{\text{alloc}_{jk}}$ denotes the allocation task for task τ_j for group γ_k , and if $\gamma_k \in \text{parents}(\gamma_l)$, then $\tau^{\text{alloc}_{jk}}$ is the parent of $\tau^{\text{alloc}_{jl}}$ (τ_j if $a_l \in \mathcal{A}$). Consequently, for a given task $\tau^{\text{alloc}_{jk}} \in \mathcal{T}_{\text{alloc}}$, we define its set of parent tasks

as: $\text{parents}(\tau^{\text{alloc}_{jk}}) = \{\tau^{\text{alloc}_{jl}} \mid \gamma_l \in \text{parents}(\gamma_k)\}$. For $\tau_j \in \mathcal{T}$, $\text{parents}(\tau_j) = \{\tau^{\text{alloc}_{jk}} \mid \gamma_k \in \mathcal{G}\}$. We then define $\text{subgroups}(\gamma)$ as the set of all subgroups included in γ .

If task $\tau_j \in \mathcal{T}$ requires allocation, an allocation instance $\tau^{\text{alloc}_{jk}}$ is created to determine the transfer of responsibility for τ_j from the parent group γ_k to one of its subgroups $\gamma_l \in \text{subgroups}(\gamma_k)$.

Finally, the proxy role is linked to m^{alloc} in DS through deontic relations of the form $\text{per}(\text{proxy}, m^{\text{alloc}}, tc)$ or $\text{obl}(\text{proxy}, m^{\text{alloc}}, tc)$, granting authority and responsibility to proxy agents to bid on behalf of their groups during distributed auctions.

5.3 Data Structures

Each agent maintains eight internal structures, identical to those defined in the I-CBBA framework [6] (as they can accommodate both CBBA and I-CBAA processes).

The *winning bid list* \mathbf{y}_i (of size $|\mathcal{T}| + |\mathcal{T}_{\text{alloc}}|$) stores the highest bid value recorded across the fleet for each task, while the *winning agent list* \mathbf{z}_i (also of size $|\mathcal{T}| + |\mathcal{T}_{\text{alloc}}|$) identifies the agent currently holding the winning bid for each task. The *timestamp list* \mathbf{s}_i (of size $|\mathcal{A}| + |\mathcal{SG}|$) contains the most recent communication time from each agent, used to resolve consensus conflicts. Each agent also keeps its local *bundle* \mathbf{b}_i , representing the ordered list of tasks currently assigned to it, and the corresponding *path* \mathbf{p}_i , defining their execution order. In addition, the *fleet bids matrix* $\mathbf{f}_i \in \mathbb{R}^{(|\mathcal{T}|+|\mathcal{T}_{\text{alloc}}|) \times (|\mathcal{A}|+|\mathcal{SG}|)}$ stores, for each pair (j, r) , the highest bid for task τ_j known by agent a_i to have been made by/for agent a_r . The associated *fleet priority matrix* ϕ_i records, for each bid in \mathbf{f}_i , the priority level of its emitter, as defined in the *priority vector* $|\mathbf{P}_\rho \in \mathbb{Z}^{|\mathcal{A}|+|\mathcal{SG}|}$, where each entry $\mathbf{P}_\rho(i)$ encodes the fixed priority level of agent a_i .

5.4 Org-CBBA Specificities

Each group $\gamma_k \in \mathcal{G}$ is represented in the bidding tables $(\mathbf{f}_i, \mathbf{y}_i, \mathbf{z}_i, \mathbf{s}_i, \phi_i)$ by a virtual entry that serves as the organizational placeholder for γ_k . Formally, if $\text{proxy} \in \pi(a_i, \gamma_k)$, then a_i intercedes on behalf of γ_k for $\tau^{\text{alloc}_{jk}}$ and is responsible to compute the corresponding bid value \mathbf{f}_{ijk} . Consequently, a task $\tau^{\text{alloc}_{jk}}$ can be assigned to a group γ , noted $\tau^{\text{alloc}_{jk}} \mapsto \gamma$.

Org-CBBA will exploit the chain of group inclusions, or *chain of affiliations*. Allocation unfolds top-down along these chains of affiliations. A concrete task $\tau_j \in \mathcal{T}$ may be allocated to an agent $a \in \mathcal{A}$ if there exists at least one path from a top-level allocation task $\tau^{\text{alloc}_{jk}}$ down to τ_j , which respect the following constraints:

- every parent group task in $\text{parents}(\tau_j)$ along the chain has been won by their respective group, and
- the deontic rules $\text{per}(\rho, m, tc)$ or $\text{obl}(\rho, m, tc)$ hold for some mission m with $\tau_j \in \text{mo}(m)$.

In other words, if multiple affiliation paths exist, the agent is permitted to allocate τ_j to itself as long as *one* of these chains is satisfied. This ensures that task responsibility is consistently delegated from higher-level groups to lower-level groups and finally to individual agents, while allowing flexibility in cases where the organizational structure provides multiple valid paths. This mechanism is therefore not hard-coded for a particular organizational shape: it follows

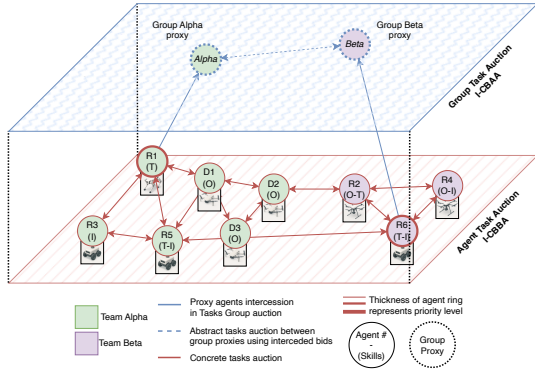


Figure 2: Illustration of Org-CBBA. At the group level, auctions for allocation tasks are voiced through proxy agents interceding on behalf of their groups. At the agent level, CBBA auctions allocate concrete tasks among individual agents.

the chains of affiliations specified in OE, thereby aligning the allocation process with the organizational structure provided, and it enables local checking of the validity of allocations, as follows:

DEFINITION 3 (ORGANIZATIONAL VALIDITY). *The allocation of a task τ to an agent a_i is organizationally valid, noted $\text{valid}(\tau, a_i)$ iff all these conditions hold:*

- (i) τ is not already completed,
- (ii) there exists ρ, m, tc, γ such that $\rho \in \pi(a_i, \gamma)$ and $\text{per}(\rho, m, tc)$ or $\text{obl}(\rho, m, tc)$ holds with $\tau \in \text{mo}(m)$, and
- (iii) if $\tau \in \mathcal{T}$, $\kappa(\tau) \subseteq \sigma(a_i)$,
- (iv) for each task $\tau^{\text{alloc}_{jk}} \in \text{parents}(\tau)$, $\tau^{\text{alloc}_{jk}} \mapsto \gamma \in \text{parents}(a_i)$

At each step of the allocation cascade, when responsibility is passed to a deeper level in the hierarchy, the winning bids are refined based on more detailed information from a smaller subset of agents. These refined values are then backpropagated upward along the affiliation chain to update the corresponding allocation task bids. If the refined information alters the outcome of any higher-level auction, the current winning branch must release the task and all its dependent tasks, and the allocation is redirected to the new winning group. This iterative correction ensures that higher-level approximations are continually aligned with progressively refined information, while preserving the fully distributed nature of CBBA.

EXAMPLE 1 (HIERARCHICAL ALLOCATION IN ZONE C). *Suppose a new observation task O_4 appears in Zone C. An allocation task A_C is created to decide whether Group Alpha or Group Beta (both $\in \mathcal{G}$) should take responsibility. Group representatives intercede on behalf of their groups to bid for A_C , using, e.g., the inverse Euclidean distance between the task location and their group centroid as a coarse proxy. If Group Alpha has the stronger centroid-based bid, it wins A_C . Only robots in Group Alpha then compete for O_4 with individual bids, subject to $\kappa(O_4) \subseteq \sigma(a)$ and permissions/obligations from DS.*

EXAMPLE 2 (REFINEMENT AND BACKPROPAGATION IN ZONE C). *Suppose Group Alpha initially wins the allocation task A_C for O_4 . Within Group Alpha, all eligible robots bid for O_4 based on their distances. The winning robot's refined bid is backpropagated upward*

to update Alpha's effective bid for A_C . If this refined value is weaker than Group Beta's original bid, the allocation of A_C switches to Beta, forcing O_4 to be reallocated accordingly.

5.5 Phase 1: Auction Process

The auction phase of Org-CBBA extends CBBA by embedding organizational constraints and handling both allocation tasks and concrete tasks. The process, summarized in Algorithm 1, unfolds as follows.

Step 1 [17, 116] – Breadth-first parsing of tasks. Agents traverse the set of goals/tasks using a breadth-first strategy (sortedBF) along OE. Only valid allocations are considered (see Definition 3).

Step 2 [18–115] – Group coordination and bidding. This step is only active when both conditions hold: (i) $\tau_j \in \mathcal{T}_{\text{alloc}}$, i.e. τ_j is a $\tau^{\text{alloc}_{jk}}$, and (ii) a_i acts as a group proxy for group $\gamma_l \in \mathcal{G}$ for which $\gamma_l \in \text{subgroups}(\gamma_k)$, i.e., proxy $\in \pi(a_i, \gamma_l)$ and $\gamma_l \in \text{subgroups}(\gamma_k)$. In this case, a_i computes a bid c_{ijl} using BID^{gp} (the bidding logic used for group tasks) on behalf of its group γ_l through bid intercession [5, 6]. The resulting bid is integrated into the fleet bid matrix \mathbf{f} using merge (19). The merge process ensures that, for each conflicting bid emitted (if multiple agents hold the role proxy), the value and priority of the highest-priority ϕ_{ii} emitter are retained, systematically prioritizing higher-priority entries (see [6, Algorithm 2]).

a_i then updates \mathbf{y} (winning bids), \mathbf{z} (winning group), and \mathbf{s} (times-tamps) based on the outcome of the auction at that group level, considering all bids currently present for τ_j . Refined subgroup bids are then backpropagated upward (Algorithm 2) so that ancestor group tasks reflect the updated valuations of their subgroups. If this refinement changes a parent's outcome, the corresponding allocation is revised and, if necessary, re-auctioned. New group tasks may also be instantiated when previously unexplored ancestors are revealed in the organizational hierarchy.

Step 3 [116] – Concrete task discovery. If τ_j is a concrete task, it is added to the available set \mathcal{T}_i for bundle-building. Marginal gain computations are restricted to this set, reducing computation compared to evaluating all tasks.

Step 4 [118–127] – Bundle-building with marginal gains. On \mathcal{T}_i , bundle construction follows the standard CBBA procedure [3]. The bid for each task $\tau_j \in \mathcal{T}_i$ is determined using $\text{BID}^{\text{ind}}(j, i)$, which, as in the original CBBA algorithm, computes for every $j \in \mathcal{T}_i \setminus \mathbf{b}_i$ the marginal gain c_{ij} of inserting τ_j into \mathbf{p}_i at the position yielding the largest increase in utility:

$$\text{BID}^{\text{ind}}(j, i) = \max_{n \leq |\mathbf{p}_i|} \left(S_i^{\mathbf{p}_i \oplus n \{j\}} - S_i^{\mathbf{p}_i} \right), \quad \forall j \in \mathcal{T}_i \setminus \mathbf{b}_i. \quad (9)$$

These values are merged into \mathbf{f} with merge. Valid tasks \mathbf{h}_i are determined by comparing with current winners, and bundle-building continues until no more tasks can be inserted.

5.6 Phase 2: Consensus Process

The consensus phase in Org-CBBA works almost exactly as in I-CBBA [6] (given we use the datastructures as defined in the I-CBBA algorithm): agents exchange and merge their local states $(\mathbf{y}_i, \mathbf{z}_i, \mathbf{s}_i, \mathbf{f}_i, \phi_i)$, update winning bids, and drop tasks if outbid. The only difference is that when a group task is dropped, any affiliated concrete tasks (and tasks following it in \mathbf{b}) already present in the agent's plan must also be dropped. This ensures that agents cannot

Algorithm 1: Org-CBBA Auction Phase for agent a_i

```

1 Procedure BUILD BUNDLE
  ( $y_i(t-1), z_i(t-1), b_i(t-1), p_i(t-1), f_i(t-1), \phi_i(t-1)$ )
2    $y_i(t) = y_i(t-1); z_i(t) = z_i(t-1)$ 
3    $b_i(t) = b_i(t-1); p_i(t) = p_i(t-1)$ 
4    $f_i(t) = f_i(t-1); \phi_i(t) = \phi_i(t-1);$ 
5    $\mathcal{T}_i \leftarrow \emptyset$ 
6   foreach  $\tau_j \in \text{sortedBF}(\mathcal{T} \cup \mathcal{T}_{\text{alloc}})$  s.t.  $\text{valid}(\tau_j, a_i)$  do
7     if  $\tau_j$  is a  $\tau^{\text{alloc}jk}$  and proxy  $\in \pi(a_i, \gamma_l)$  and  $\gamma_l \in \text{subgroups}(\gamma_k)$ 
8       then
9          $c_{ijl} \leftarrow \text{BID}^{\text{EP}}(j, l)$ 
10        merge( $f_{ijl}(t), c_{ijl}, \phi_{ijl}(t), P_\rho(i)$ )
11        if  $\max(f_{ij}(t)) > y_{ij}$  then
12           $y_{ij}(t) \leftarrow \max(f_{ij}(t))$ 
13           $z_{ij}(t) \leftarrow \text{winning representative}$ 
14           $s_{ij} \leftarrow \text{current timestamp}$ 
15          backpropagate( $\tau_j, \max(f_{ij}(t))$ )
16          create_group_task( $\tau_j$ )
17        else if  $\tau_j$  is a concrete task then  $\mathcal{T}_i \leftarrow \mathcal{T}_i \cup \{\tau_j\};$ 
18  while  $|\mathcal{T}_i \setminus b_i| > 0$  do
19     $c_{iji} = \text{BID}^{\text{ind}}(j, i)$ 
20    merge( $f_{iji}(t), c_{ij}, \phi_{iji}(t), P_\rho(i)$ ),  $\forall j \in \mathcal{T}_i \setminus b_i$ 
21     $h_{ij} = \mathbb{I}(f_{iji}(t) > y_{ij}), \forall j \in \mathcal{T}_i \setminus b_i$ 
22    if  $|h_i| = 0$  then terminate process // no valid tasks;
23     $J_i = \text{argmax}_j h_{ij} \cdot f_{iji}(t)$ 
24     $n_{iJ_i} = \text{argmax}_n S_i^{p_i \oplus n(J_i)}$ 
25     $b_i \leftarrow b_i \oplus_{\text{end}} \{J_i\}$ 
26     $p_i \leftarrow p_i \oplus_{n_{iJ_i}} \{J_i\}$ 
27     $y_{iJ_i}(t) \leftarrow f_{iJ_i}(t)$ 
28     $z_{iJ_i}(t) \leftarrow i$ 
29    backpropagate( $J_i, f_{iJ_i}(t)$ )

```

Algorithm 2: Backpropagation for task τ_j and value c

```

1 Procedure backpropagate( $\tau_k, c$ )
2   foreach parent  $\tau_k \in \text{parents}(\tau_j)$  do
3     update  $f_{ik}(t)$  with  $c$ 
4     if winning bid at  $\tau_k$  changes then
5       update  $y_{ik}(t), z_{ik}(t)$ 
6       if  $\tau_k \in b_i$  then
7         foreach  $\tau_\ell \in \{\text{tasks following } \tau_k \text{ in } b_i\}$  do
8            $b_i \leftarrow b_i \setminus \{\tau_\ell\}$ 
9            $p_i \leftarrow p_i \setminus \{\tau_\ell\}$ 
10           $y_{i\ell} \leftarrow 0$ 
11           $z_{i\ell} \leftarrow \emptyset$ 
12    backpropagate( $\tau_k, c$ )

```

retain concrete tasks whose parent group task is no longer held, while leaving all other consensus mechanics unchanged.

5.7 Convergence and Optimality Guarantees

Org-CBBA inherits the convergence properties of its constituent mechanisms [3, section V-D]. At each branching point: (i) tasks are first allocated to groups via CBAA, which guarantees finite-time convergence to a unique winner, and (ii) within the winning group, CBBA is applied to allocate concrete tasks to individual agents, also guaranteeing convergence to a conflict-free allocation. Since these processes are chained hierarchically, the overall system converges to a consistent allocation in which each task is assigned to exactly one agent.

Bid backpropagation adjusts bid valuations without altering the consensus mechanics of CBAA/CBBA and thus does not affect convergence. For the mechanism to operate correctly, bids at group and agent levels must remain *comparable in magnitude*. If group-level bids are underestimated, backpropagation may fail to trigger needed reallocations; if agent-level bids are too small, every refinement would overturn higher-level results, collapsing the hierarchy into a flat CBBA. Hence, group-level bids should be *optimistic*—slightly overvaluing their agents’ expected performance—so that backpropagation only corrects significant discrepancies. As in CBBA, Org-CBBA ensures convergence but not global optimality, producing locally optimal allocations consistent with OE.

6 EXPERIMENTAL EVALUATION

6.1 Driving scenario

Consider an initial setup where the 9 robots depart from the central base. Early in the mission, several tasks appear in Area A: for example, an observation task O_1 , a trapping task T_1 , and an interception task I_1 . The robots eligible for each task bid according to their inverse Manhattan distance. For instance, O_1 can be claimed by any of the six robots possessing the O-skill, T_1 by any of the four robots with T-skill, and I_1 by any of the four robots with I-skill. The closest eligible robots win their respective bids, illustrating how spatial proximity and capability constraints jointly determine allocation outcomes in this motivating scenario.

6.2 Experimental Setup

To evaluate Org-CBBA, we compare two organizations, each representing a different way of structuring the decision process.

Flat organization (CBBA baseline). The fleet operates without hierarchy: every agent $a_i \in \mathcal{A}$ is considered for every task $\tau_j \in \mathcal{T}$ such that $\kappa(\tau_j) \subseteq \sigma(a_i)$. The bidding function computes the marginal cost of inserting a task into the current plan, using the *inverse of the shortest path length* as a distance-based utility, whilst meeting skill requirements. This corresponds to a direct application of the standard CBBA framework.

Hard hierarchy. The fleet is organized into two heterogeneous groups: Alpha (six robots: 3×O, 1×T, 1×I, 1×T,I) and Beta (three robots: 1×O,T, 1×O,I, 1×T,I). Group *Alpha* is exclusively responsible for tasks in Zone A, while Group *Beta* handles those in Zone B; neither group can handle tasks in the other’s zone. This exclusivity is enforced through a logic-gate rule: if a task lies in Zone A, Alpha’s bid is set to 1 and Beta’s to 0, and symmetrically for Zone B. Both groups compete for each new concrete task τ_j appearing in the shared Zone C. Group-level bids for these allocation tasks are computed as the *inverse Euclidean distance* between the task location and the nearest agent of each group. The winning group then secures responsibility for the corresponding concrete task, which is subsequently allocated among agents at the robot level with a compatible skill set.

The mission environment is modeled as a 20×20 grid (400 nodes) with 70% of possible edges active while preserving graph connectivity. The home base is located at (10, 10). Each simulation lasts 200 time steps, during which tasks appear dynamically.

The primary experimental variable is the *task spawning rate in Zone A*, which determines the relative workload distribution.

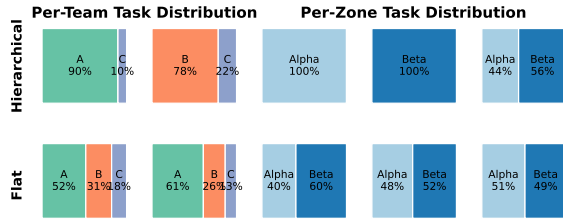


Figure 3: Task allocation distribution across organization type by groups and zones

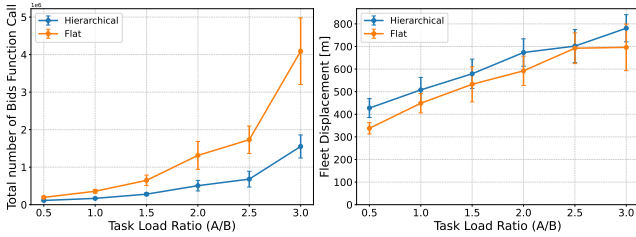


Figure 4: Bid count and total displacement vs ratio of task load in zone A and B

Table 1: Difference in cumulated displacement (across the fleet) and bid calls between hierarchical and flat configurations across task zone ratios.

Task Zone Ratio	Fleet Displacements (hierarchical – flat)	Bid Computations (hierarchical – flat)
0.5	89.50 (26.48%)	-82775.80 (-42.26%)
1	59.00 (13.14%)	-189356.60 (-52.95%)
1.5	46.60 (8.75%)	-367818.80 (-56.71%)
2	81.10 (13.69%)	-806959.50 (-61.42%)
2.5	9.30 (1.34%)	-1050260.60 (-60.66%)
3	84.50 (12.14%)	-2539189.60 (-62.07%)

Tasks in Zone B are generated at a constant rate across runs, while the rate in Zone A is progressively increased. Zone C maintains a constant task count. Varying the Zone A rate produces different task load ratios (A/B), enabling systematic comparison of organizational structures under distinct spatial workload distributions.

For each configuration, task arrival timelines are generated with randomized release times to capture deployment variability. Ten scenarios are simulated per case for statistical robustness.

Performance is evaluated along three dimensions: (i) *Cumulative fleet displacement*: total distance traveled by all agents, serving as an optimality metric. (ii) *Bid computations*: number of calls to each bidding logic, capturing computational overhead. (iii) *Per-zone dispatch*: fraction of tasks from Zones A, B, and C allocated to each group, measuring compliance with organizational responsibilities. We hypothesize that hierarchy reduces overhead by replacing numerous fine-grained bids with abstract group-level bids, progressively constraining the solution space while preserving allocation quality and organizational validity.

6.3 Results Analysis

The aggregated results across all task load ratios are shown in Figure 3. Values correspond to mean performance over ten randomized scenarios per configuration.

In Figure 3, the *Hierarchical* configuration enforces organizational constraints as designed: Team Alpha handles Zone A, Team Beta handles Zone B, and both share Zone C via group-level bidding. A slight asymmetry in Zone C favors Beta, explained by the higher task density in Zone A, which increases Alpha’s workload and reduces its competitiveness in shared tasks. By contrast, the *Flat* configuration exhibits significant cross-zone mixing, as both teams compete across all zones without structural separation.

As reported in Table 1, the Hierarchical configuration incurs a moderate increase in cumulative fleet displacement relative to the Flat baseline (+1.3% to +26.5%), reflecting the limited optimality loss induced by zone exclusivity. However, bid evaluations decrease by 42–62%, confirming a substantial reduction in communication and decision-making load through group-level abstraction. As the task load in Zone A increases, computational savings grow while the displacement gap stabilizes below 15%, indicating that hierarchical decomposition becomes increasingly beneficial under spatially structured workloads.

Overall, introducing organizational structure ensures strict zone fidelity and markedly improves computational efficiency, with only limited degradation in allocation quality.

7 CONCLUSIONS

This paper introduced the Org-MRTA framework, which formalizes the trade-off between task allocation efficiency and organizational validity, ensuring that multi-robot systems operate within predefined structural and normative constraints. Such alignment is critical in domains where compliance with roles, missions, and authority relations is essential, including civil protection, military, and first-response operations.

To solve Org-MRTA in a distributed manner, we proposed Org-CBBA, which leverages the organizational structure to establish consensus between groups prior to consensus within the selected group. This hierarchical consensus mechanism was implemented and evaluated on a scenario derived from the <anonymized> challenge, designed to stress mission execution under unpredictable task arrivals. Compared to a flat CBBA without organizational structure, Org-CBBA achieves efficient allocations with an average optimality gap below 13%, while satisfying organizational constraints and significantly reducing bidding complexity, yielding more than 56% fewer bid computations.

Future work will focus on softening organizational constraints by allowing increased flexibility for robots to adapt under overload conditions, particularly when alternative groups are better suited to handle dense, unforeseen task arrivals. We also plan to evaluate Org-CBBA at larger scales, including the Robocup Rescue challenge [13], and to deploy it on a real robotic platform at ONERA for participation in the CoHoMa challenge.

ACKNOWLEDGMENTS

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

REFERENCES

- [1] Matthew E. Argyle, David W. Casbeer, and Randal W. Beard. 2011. A Multi-Team Extension of the Consensus-Based Bundle Algorithm. *Proceedings of the 2011 American Control Conference* (2011), 5376–5381. <https://api.semanticscholar.org/CorpusID:29143274>
- [2] Hamza Chakraa, François Guérin, Edouard Leclercq, and Dimitri Lefebvre. 2023. Optimization techniques for Multi-Robot Task Allocation problems: Review on the state-of-the-art. *Robotics and Autonomous Systems* 168 (2023), 104492. <https://doi.org/10.1016/j.robot.2023.104492>
- [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] 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>
- [5] Victor Guillet, Christophe Grand, Charles Lesire, and Gauthier Picard. 2025. Bid Intercession to Unlock Human Control in Decentralized Consensus-Based Multi-robot Task Allocation Algorithms. In *Agents and Robots for reliable Engineered Autonomy*, Angelo Ferrando and Rafael C. Cardoso (Eds.). Springer Nature Switzerland, Cham, 99–114.
- [6] Victor Guillet, Charles Lesire, Gauthier Picard, and Christophe Grand. 2025. Extending Consensus-based Task Allocation Algorithms with Bid Intercession to Foster Mixed-Initiative. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS-25)*, IFAAMAS, 932–940. <https://www.ifaamas.org/Proceedings/aamas2025/pdfs/p932.pdf>
- [7] Jomi Fred Hübner, Jaime Simão Sichman, and Olivier Boissier. 2002. A Model for the Structural, Functional, and Deontic Specification of Organizations in Multiagent Systems. In *Brazilian Symposium on Artificial Intelligence*. <https://api.semanticscholar.org/CorpusID:7524187>
- [8] Simon Hunt, Qinggang Meng, Chris J. Hinde, and Tingwen Huang. 2014. A Consensus-Based Grouping Algorithm for Multi-agent Cooperative Task Allocation with Complex Requirements. *Cognitive Computation* 6 (2014), 338 – 350. <https://api.semanticscholar.org/CorpusID:2626048>
- [9] Darren Hurley-Smith, Jodie Wetherall, Stephen R. Woodhead, and Andrew Adewunmi Adekunle. 2014. A Cluster-Based Approach to Consensus Based Distributed Task Allocation. *2014 22nd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing* (2014), 428–431. <https://api.semanticscholar.org/CorpusID:11026569>
- [10] Jie Jiang, Olivier Boissier, Virginia Dignum, and Javier Vázquez-Salceda. 2011. An Agent-Based Inter-Organizational Collaboration Framework based on OperA. In *In "Multiagent Systems and Applications / Workshops", Lecture Notes in Artificial Intelligence*. Springer, 58–74.
- [11] 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>
- [12] 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>
- [13] RoboCupRescue. 2025. RoboCupRescue Robot League. Retrieved October 6, 2025 from <https://rrl.roboocup.org/>
- [14] Javier Vázquez-Salceda and Virginia Dignum. 2003. HarmonIA: A New Approach to Model Electronic Institutions. In *The Role of Norms and Electronic Institutions in Multi-Agent Systems*, Javier Vázquez-Salceda (Ed.). IOS/Springer, 91–114.
- [15] Javier Vázquez-Salceda, Virginia Dignum, and Olivier Boissier. 2003. The Role of Norms and Electronic Institutions in Multi-Agent Systems. In *Proceedings of the 2nd European Conference on Artificial Intelligence for Electronic Agents (EAAI or similar)*. IOS Press or Springer, 59–88. Also in “Electronic Institutions” / journal or book chapter.
- [16] M. Weck and R. Dammertz. 1995. OPERA – A New Approach to Robot Programming. *CIRP Annals* 44, 1 (1995), 389–392. [https://doi.org/10.1016/S0007-8506\(07\)62348-8](https://doi.org/10.1016/S0007-8506(07)62348-8)
- [17] Andrew K. Whitten, Han-Lim Choi, Luke B. Johnson, and Jonathan P. How. 2022. MIT Open Access Articles Decentralized Task Allocation with Coupled Constraints in Complex Missions. <https://api.semanticscholar.org/CorpusID:8649938>
- [18] Michael Wooldridge, Franco Zambonelli, and Nicholas R. Jennings. 2000. The Gaia methodology for Agent-oriented analysis and design. *Autonomous Agents and Multi-Agent Systems* 3, 3 (2000), 285–312. <https://doi.org/10.1023/A:1010071910869>