



**HAL**  
open science

# Market-based Multi-robot coordination with HTN planning

Antoine Milot, Estelle Chauveau, Simon Lacroix, Charles Lesire

► **To cite this version:**

Antoine Milot, Estelle Chauveau, Simon Lacroix, Charles Lesire. Market-based Multi-robot coordination with HTN planning. International Conference on Intelligent Robots and Systems (IROS 2021), Sep 2021, Prague, Czech Republic. hal-03418176

**HAL Id: hal-03418176**

**<https://laas.hal.science/hal-03418176>**

Submitted on 6 Nov 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Market-based Multi-robot coordination with HTN planning

Antoine Milot<sup>1,2,3</sup>, Estelle Chauveau<sup>2</sup>, Simon Lacroix<sup>1</sup> and Charles Lesire<sup>3</sup>

**Abstract**—We propose a decentralized approach that simultaneously allocates and decomposes high level tasks among various robots. The approach exploits HTN structures and algorithms, that are used within an auction-based allocation scheme, and aims at dealing with complex tasks with causal or temporal relations. The paper formalizes the approach, and depicts how HTN planning processes are used to estimate bids and distribute tasks. Results on a statistical series of coverage problems are presented and their performance is assessed through a comparison with a state of the art algorithm.

## I. INTRODUCTION

A key ingredient in multi-robot systems is the ability to distribute the tasks to achieve within the robots. When the problem is not trivially solved, *e.g.* for small sets of robots and tasks or if robots and tasks are typed so that a distribution is readily defined, one must solve the Multi-Robot Task Allocation (MRTA) problem, for which the literature has proposed a whole corpus [1]. Given a set of  $n$  robot  $R = (r_1, \dots, r_n)$  and a set of  $s$  tasks  $Q = (q_1, \dots, q_s)$ , solving the MRTA problem consists in finding an allocation  $A : Q \rightarrow R$ , *i.e.* allocate each task  $q \in Q$  to a robot  $r \in R$ . But if the mission to accomplish is expressed at a rather high level, and not directly as a series of tasks, it has to be decomposed into tasks, sub-tasks, down to elementary actions. This decomposition can typically be achieved by a planner, that optimises some criteria while satisfying the problem constraints. The task allocation and decomposition problems are clearly not independent, and both decompose-then-allocate and allocate-then-decompose methods have drawbacks and yield sub-optimal solutions [2].

We propose in this paper an approach to solve both the planning and allocation problems in a unified manner. As a supporting context, we consider the problem of naval mine-hunting, in which a fleet of autonomous underwater vehicles (AUVs) is tasked to localize, identify and neutralize mines to secure a given area or channel. If some accomplished works exist regarding the trajectory optimisation of one AUV [3], [4], multi-robot technologies applied to minehunting are still in an exploratory phase [5], and improvements in intelligent embedded signal processing [6], [7] give meaning to this research area.

While the initial mission planning phase amounts to a multi-robot coverage problem, this operational context brings

various dynamic contingencies, as the system needs to cope with execution failures and allocation of new tasks. These unforeseen events come directly from the mission definition, during which mine identification and neutralization tasks will only be triggered at execution time by a mine detection for instance, and also from the mission environment. Besides, underwater communications are severely constrained [8]: this may imply the establishment of communication relays, for example to transmit large data to the operators for identification purposes, which introduces temporal and causal constraints in the mission. Importantly, it precludes the use of a centralized approach to ensure the proper execution of a mission. Furthermore, the underwater environment yields uncertainties in task execution, for instance underwater currents are difficult to precisely predict, whereas they significantly impact robots motions. All these elements cannot be initially accounted for: they require repair procedures that must be executed online, and which may call for a revision of the initial allocations.

Our approach integrates two well known allocation and planning paradigms: market-based allocation (MBA) [9], and Hierarchical Task Network (HTN) planning [10] (Fig. 1).

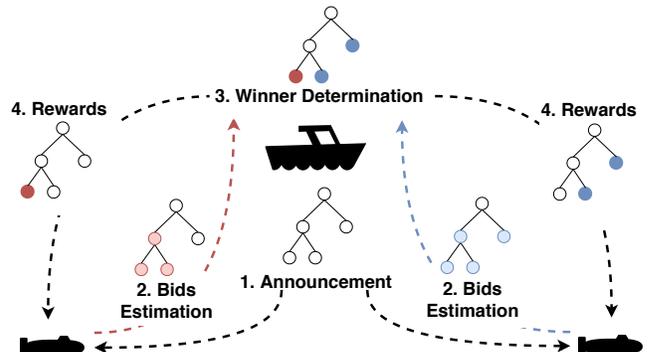


Fig. 1. Overview of the approach. The whole mission decomposed as a HTN is auctioned. Each robot bids on the sub-tasks, a winner determination process allocates the tasks and sends them to the robots (“rewards”).

MBA approaches rely on auctions: an auctioneer issues tasks to be allocated, and robots reply with bids that express the utility for them to execute the tasks. The auctioneer then allocates tasks to robots through solving a Winner Determination Problem (WDP). This approach is a solution of choice for underwater contexts, as it is naturally decentralized and offers a common way to handle MRTA problems in communication restricted environments [11], [12]. Communications with the auctioneer are only required during auctioning phases, and the system is designed such that any robot can become an auctioneer. This feature is a

<sup>1</sup>LAAS-CNRS, Université de Toulouse, CNRS 7, Avenue du Colonel Roche, 31031 Toulouse, France antoine.milot@laas.fr simon.lacroix@laas.fr

<sup>2</sup>NAVAL GROUP, 99 Avenue Pierre-Gilles de Gennes, 83190 Ollioules, France estelle.chauveau@naval-group.com

<sup>3</sup>ONERA/DTIS, Université de Toulouse, 2 avenue Edouard Belin 31055 Toulouse, France charles.lesire@onera.fr

very flexible way to repair and update plans: for example a robot can become an auctioneer to resell tasks that it can no longer achieve, or to handle new tasks to achieve (e.g. a detected mine to be identified).

HTN planning is a major research field in planning, whose core concept is to reason with *primitive tasks* and *compound tasks*. Primitive tasks are tasks that can be directly executed by agents, and compound tasks can be broken down into others compound or primitive tasks, and their definition encodes expert knowledge on the problem domain [10]. In our approach, HTN planning is used for two distinct purposes during the MBA protocol: for robots to define their bids, and more interestingly for the auctioneer to solve the WDP. Indeed, posing the WDP as a planning problem allows to integrate task dependencies, such as temporal and causal constraints, which can be handled with an HTN approach, thanks the expressivity of the problem representation and the efficacy of HTN solvers [13], [14].

By integrating these two allocation and planning paradigms, our ambition is to lay down the foundations of a principled approach that includes task dependencies and constraints to solve complex multi-robot mission planning and execution supervision problems. In this paper, we formalize the problem and describe its cornerstone, i.e. how to solve one auction with HTN planning. The next section reviews the related work, and section III introduces the basic concepts of auction-based allocation and HTN planning. Section IV is the core of the paper: it introduces how the bidding and winner determination processes of an auction scheme are formalised and solved as HTN planning problems. Section V presents and analyses a series of results.

## II. RELATED WORK

Market-based approaches to solve the MRTA problem have been studied for long [9]. Noteworthy, these approaches are naturally decentralized and hence allow to handle MRTA problems in communication restricted environments [11], [12]. In classic MRTA problems, a set of simple tasks has to be allocated. Traditional auction-based approaches handle these cases with good performances [15], [16], [17]. These schemes usually do not consider task dependencies and constraints. Moreover, when dealing with complex tasks having more than one way to be accomplished, i.e. when several solutions solve the problem, their plan quality is sorely dependant on the order in which tasks are allocated.

Some work focused on integrating precedences or time windows constraints. In [18], temporal constraints are dealt with a global Simple Temporal Network (STN) and an iterated auction scheme. However, because their approach uses a fixed plan, it is unable to handle complex tasks.

On the other hand, to cope with more complex MRTA problems with auction-based approaches, the auctioneer, by reasoning on hierarchically linked tasks, can choose and sell parts of a plan. For example in [19], which is a reference work on merging auction schemes and hierarchically linked tasks, items for sale are replaced by task trees: hierarchical constraints are represented with *AND* and *OR* nodes, that

allows complex tasks to be decomposed into less complex ones. A direct benefit is to interleave decomposition and allocation, a key element to get better performances while solving a complex MRTA problems. Indeed, by selling parts of a task tree, auction schemes can deal with the expression of dependencies between tasks, reason on plans to solve a complex problem, and simultaneously sell several tasks at once. Leveraging on [19], an architecture based on auctions on task trees, while implementing the possibility for local bidders to buy a task for other teammates, is presented in [20]. In the same vein, [21] integrated the possibility to resell a task if the winner finds a better decomposition.

In these works, the WDP solution is found in a greedy manner, by running a breadth-first search algorithm on the best offers received. This efficiently produces good solutions, but their quality depends on the order in which nodes are visited. Moreover, these algorithms do not fully tackle dependencies and constraints between tasks. In [20], the number of temporal constraints over a node is used to order the allocation process but cross-schedules dependencies induced by these temporal constraints are never checked, while [19], [21] only suggest some perspectives relying on consensus and opportunistic subteam formation.

We believe that a greedy approach for solving WDP lacks sufficient flexibility to integrate orderings, causal, or temporal constraints between tasks efficiently. However, we see in HTN planning, which is traditionally used for satisfaction problems but is also able to handle cost optimization and temporal constraints [10], an opportunity to handle these drawbacks. Therefore, we propose to formally define in this paper the basis of an approach exploiting HTN planning for auction-based allocation solutions.

## III. BACKGROUND

### A. Basics of market-based allocation

Auctions allow items exchange between agents through bids issuing. To produce these bids, each bidder estimates for each auctioned item a cost<sup>1</sup>, that are afterwards compared by the auctioneer which allocates tasks to agents by maximizing a specified utility criteria. The most basic scheme is the Single-Item (SI) auction [22]: in this scheme, only one item is put for sale by the auctioneer. Each bidder produces a single bid, and the winner determination is usually straightforward: the best bid wins. The steps of this scheme are (Fig. 1): 1) **Announcement**: the auctioneer opens an auction by broadcasting the information on the item for sale; 2) **Bids estimation**: each bidder estimates the cost associated to the item and sends its bid to the auctioneer; 3) **Winner Determination**: once the auctioneer received all bids, or a deadline is reached, it resolves the WDP to find an allocation; 4) **Reward**: if a winner has been found at the previous step, the auctioneer sends him the tasks it has to execute.

Numerous variations of this scheme can be defined, depending on the nature and number of the items that are traded, the bidding policy, or the criteria optimized by the

<sup>1</sup>or a reward or a utility – we use costs throughout the paper

WDP. In our work, the allocation scheme is based on Sequential Single-Item (SSI) auctions. The main difference with respect to SI is that auctions are lists of items: each bidder produces a bid for each item of the list. If after solving the WDP some items remain not allocated, the auctioneer starts a new round. The process goes on until all items are allocated or a stop criterion is reached. In order to ensure bids independence, a bidder can win at most one item at each round. SSI allows robots to prioritize some items over others and can speed up the overall allocation process, but has some pitfalls [22]: the WDP complexity increases exponentially with the number of items and bids, and it is difficult to express bids dependencies. However, these two points can be addressed by auctioning over task trees [19]: by bidding on a subtree, bidders can express some kind of dependencies on the tasks it contains, and from one perspective, these subtrees are a constrained form of combinatorial solutions allowing to avoid the computational bottleneck.

### B. HTN planning

We recall here the basic concepts of HTN planning [23], [24]. First, we define a first-order language  $\mathcal{L}$  that contains symbols used in HTN planning.  $\mathcal{L}$  is a tuple  $(\mathbb{V}, \mathbb{C}, \mathbb{P}, \mathbb{T}_P, \mathbb{T}_C, \mathbb{L})$  where  $\mathbb{V}$  is an infinite set of variable symbols,  $\mathbb{C}$  is a finite set of constants symbols,  $\mathbb{P}$  is a finite set of predicate symbols,  $\mathbb{T}_P$  is a finite set of primitive task symbols,  $\mathbb{T}_C$  is a finite set of compound task symbols, and  $\mathbb{L}$  is an infinite set of labels. A predicate  $p \in \mathbb{P}$  is said *ground* if all its parameters are constants of  $\mathbb{C}$ . From this language, we can then define tasks:

**Definition 1** (Tasks). Given a set of terms  $x_1, \dots, x_k$  issued from  $\mathcal{L}$ , and  $t \in (\mathbb{T}_P \cup \mathbb{T}_C)$  a task symbol, then  $t(x_1, \dots, x_k)$  is a *task*, i.e. an instance of  $t$  on parameters  $x_1, \dots, x_k$ .

In HTN planning, primitive tasks are not decomposable, and compound tasks can be decomposed into other tasks through *methods*. Both primitive and compound tasks can have *preconditions*, i.e. a first-order formula on  $\mathcal{L}$ , but only primitive tasks can change the world by applying effects, denoted by literals over  $\mathcal{L}$  predicates. The decomposition of compound tasks is represented by partially ordered sub-plans, named *task networks*. As tasks can appear several times in a plan (e.g., a move to a same position by a same robot), we need to *label* each occurrence of a task  $t(x_1, \dots, x_k)$  in the plan with a label  $l \in \mathbb{L}$ .

**Definition 2** (Task network). A task network  $tn$  over a set of tasks  $X$  is a tuple  $(L, \prec, \alpha)$  with the following elements:

- 1)  $L \subset \mathbb{L}$  is a set of labels
- 2)  $\prec$  is a strict partial order over  $L$
- 3)  $\alpha : L \rightarrow X$  maps labels to tasks

A task network is *grounded* if its elements contain no variable. A task network only composed of primitive tasks is called a *primitive task network*.

**Definition 3** (Decomposition Method). A decomposition method  $m$  is a pair  $(t_c, tn)$  with  $t_c$  a compound task and

$tn$  a task network. It states that  $t_c$  can be refined by  $tn$ , i.e. that one way of achieving  $t_c$  is to achieve the tasks of  $tn$ .

We can now define two last concepts of HTN planning: the *planning domain* and *planning problem*.

**Definition 4** (Planning Domain). A planning domain  $\mathcal{D}$  is a tuple  $(\mathcal{L}, T_P, T_C, M)$  with:

- $\mathcal{L}$  the underlying HTN language
- $T_P$  and  $T_C$  sets of primitive and compound tasks
- $M$  a set of decomposition methods s.t.  $\forall m = (t_c, tn) \in M, t_c \in T_C$  and  $tn$  is a task network over  $T_P \cup T_C$ .

**Definition 5** (Planning Problem). A planning problem  $\mathcal{P}$  is a tuple  $(\mathcal{D}, s_I, tn_I)$  where:

- $\mathcal{D}$  is a planning domain;
- $s_I$  is the initial state, defined as conjunction of ground literals of  $\mathcal{L}$ ;
- $tn_I$  is the initial task network.

Solving an HTN problem  $\mathcal{P} = (\mathcal{D}, s_I, tn_I)$  consists in finding a *solution* task network  $tn$  such that  $tn$  is *primitive* and *executable* in  $s_I$ , i.e., there is a sequence of  $tn$  tasks, that respects the ordering constraints, in which the preconditions of a task are valid in the state resulting from applying the previous task. HTN solving algorithms generally iteratively build  $tn$  by exploring the possible decomposition methods that can be applied from  $tn_I$  [25], [10].

## IV. AUCTIONING HTN PLANS

Our approach is built upon SSI auctions using HTN structures as items to trade. Bidders estimate bids on HTN tasks, using HTN planning. The auctioneer allocates subtrees of the tasks network, ensuring that no nodes are allocated more than once, by solving the WDP using HTN planning.

### A. Auction definition

Items announced by the auctioneer represent *tasks* on which each robot bids. All the robots involved in the auction must share a *common* knowledge about the tasks that will be possibly announced. In addition, this common knowledge will be combined by the robots with their own *local* knowledge representing their local capacities.

The common knowledge is embodied in a common HTN domain  $\mathcal{D}_{com} = (\mathcal{L}_{com}, \emptyset, T_{C_{com}}, M_{com})$ , with  $\mathcal{L}_{com}$  a common language defining variables, constants and symbols known by all robots,  $T_{C_{com}}$  the common compound tasks (note that there is no common primitive tasks), and  $M_{com}$  the methods that decompose the common tasks.

We rely on HDDL [24] in order to practically represent the common domain. Listing 1 shows an excerpt of the common domain in HDDL for a coverage mission.  $\mathcal{L}_{com}$  is defined by types, predicates, variables... The tasks that can be allocated to the robots are *cover* tasks, which can be decomposed into other *cover* sub-tasks using methods *m-cover-d*.

Note that from the point of view of HTN planning,  $\mathcal{L}$  is fully known, i.e. all elements are declared before planning. While it is reasonable for most elements in  $\mathcal{L}$  (variables, predicates, ...), it must be revised when dealing

```

(:types area - task)
(:predicates (area-decomposition ?a ?a1 ?a2 - area))
(:task cover :parameters (?a - area))
(:method m-cover-d
:parameters (?a ?a1 ?a2 - area)
:task (cover ?a)
:precondition (area-decomposition ?a ?a1 ?a2)
:subtasks (and (cover ?a1) (cover ?a2)))

```

Listing 1. Excerpt of the HDDL description of a coverage mission  $\mathcal{D}_{com}$

with *constants*, that represent the considered objects. If these objects can be considered known when performing a single auction, they can evolve during the mission (e.g., new objects discovered, new areas to cover specified by the operator). The language constants then cannot be fully defined in  $\mathcal{D}_{com}$ , and some constants may be sent when announcing a new auction. When new tasks have to be allocated, the auctioneer sends an announcement message that contains these tasks. We propose to structure this message through a *grounded HTN tree*.

**Definition 6** (Grounded HTN Tree). Let  $\mathcal{P} = (\mathcal{D}, s_I, tn_I)$  a planning problem with  $\mathcal{D} = (\mathcal{L}, T_P, T_C, M)$  the associated domain. Without loss of generality, we assume that  $tn_I$  contains just a single ground compound task  $top$  with label  $l_{top} \in \mathbb{L}$ , for which there is exactly one method in  $M$ . The bipartite tree  $\mathcal{H} = (V_T, V_M, E_{T \rightarrow M}, E_{M \rightarrow T})$ , consisting of labeled task vertices  $V_T$ , labeled method vertices  $V_M$ , and edges  $E_{T \rightarrow M}$  and  $E_{M \rightarrow T}$  is a grounded HTN tree if:

- 1)  $v_{top} = (l_{top}, top) \in V_T$
- 2) Let  $v_k = (l_k, t_k) \in V_T$ , with  $l_k \in \mathbb{L}, t_k \in T_C$ , then  $\forall m \in M$  s.t.  $m = (t_k, tn)$  holds:
  - $v_m = (l_k, m) \in V_M$
  - $(v_k, v_m) \in E_{T \rightarrow M}$
- 3) Let  $v_k = (l_k, m_k) \in V_M$ , with  $l_k \in \mathbb{L}, m_k = (t_k, tn_k), t_k \in T_C$ , and  $tn_k = (L_k, \prec_k, \alpha_k)$  a task network. Then  $\forall l_j \in L_k$  holds:
  - $v_t = (l_{k,j}, \alpha_k(l_j)) \in V_T$
  - $(v_m, v_t) \in E_{M \rightarrow T}$
- 4)  $\mathcal{H}$  is minimal, such that 1 to 3 hold.

For this grounded HTN tree  $\mathcal{H}$  to be finite (and then be buildable), we must ensure that the HTN problem is *acyclic*.<sup>2</sup> In order to build the grounded HTN tree  $\mathcal{H}$  that defines the tasks to allocate and their hierarchical decomposition, the auctioneer relies on both the common domain  $\mathcal{D}_{com}$ , and a planning problem corresponding to the current auction. Listing 2 shows an excerpt of the HDDL model of such a planning problem in which one top area  $a-1$  is further decomposed into a set of sub-areas to cover. The HTN tree  $\mathcal{H}$  corresponding to this problem is shown in Fig. 2.

From this grounded HTN structure, we can define the elements that constitute the *item for sale*.

**Definition 7** (Item for sale). An item for sale  $\delta$  is a tuple  $(\mathcal{H}_\delta, C_\delta, s_\delta, L_\delta)$ , with:

<sup>2</sup>It is possible to verify that  $\mathcal{H}$  is finite by first building the problem Task Decomposition Graph (TDG), which is always finite, and then checking that the TDG is acyclic [14].

```

(:objects
a-1 a-1-1 a-1-2 a-1-2-d1 a-1-2-d2 - area
a-1-2-d1-1 a-1-2-d1-2 a-1-2-d2-1 a-1-2-d2-2 - area)
(:htn :tasks (cover a-1))
(:init
(area-decomposition a-1 a-1-1 a-1-2-d1)
(area-decomposition a-1 a-1-1 a-1-2-d2)
(area-decomposition a-1-2-d1 a-1-2-d1-1 a-1-2-d1-2)
(area-decomposition a-1-2-d2 a-1-2-d2-1 a-1-2-d2-2))

```

Listing 2. Excerpt of the HDDL problem describing an auction.

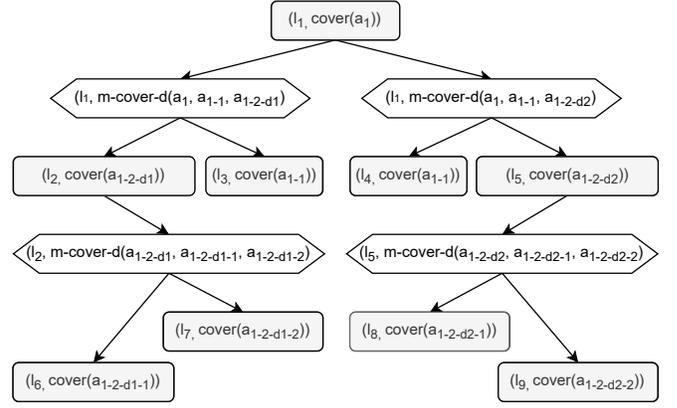


Fig. 2. Grounded HTN tree for the problem listed in Listing 2. Rounded rectangles represent labeled tasks (in  $V_T$ ), hexagons represent labeled methods (in  $V_M$ ).

- 1)  $\mathcal{H}_\delta$  is a finite grounded HTN tree representing the decomposition constraints between tasks;
- 2)  $C_\delta$  contains language constants for this auction;
- 3)  $s_\delta$  is a set of atomic formulas on these constants;
- 4)  $L_\delta$  is the set of task labels in  $\mathcal{H}_\delta$  that are *sellable*, i.e. on which robots can produce bids.

### B. Bid estimation

To benefit from the HTN model that is embodied both in the common domain  $\mathcal{D}_{com}$  and the HTN tree  $\mathcal{H}_\delta$  of the items, bids are estimated using an HTN planner. This estimation is performed in two steps: first, a *bid domain* is built from the common domain completed by  $\mathcal{D}_{loc} = (\mathcal{L}_{loc}, T_{P_{loc}}, T_{C_{loc}}, M_{loc})$  a local domain defining language, primitive tasks, compound tasks and methods proper to the bidder; then, for each sellable task, a *bid estimation problem* is defined and solved.

For example, in the coverage mission with the auction defined by the HTN of Fig. 2, each robot bids on the *cover* tasks. The local domain encapsulates the decomposition of these tasks, with respect to the basic actions that can be performed by the robot (e.g., the coverage pattern may differ depending on the robot sensors). In particular, the local domain contains as primitive tasks the basic *actions* that can be performed by the bidder and that will increment the cost of the plan.

Each bidder builds its bid domain  $\mathcal{D}_{bid}$  by extending the common domain  $\mathcal{D}_{com}$  with its own local domain  $\mathcal{D}_{loc}$ :

$$\mathcal{D}_{bid} = (\mathcal{L}_{com} \cup \mathcal{L}_{loc}, T_{P_{loc}}, T_{C_{com}} \cup T_{C_{loc}}, M_{com} \cup M_{loc}) \quad (1)$$

with  $T_{C_{com}} \cap (T_{C_{loc}} \cup T_{P_{loc}}) \neq \emptyset$ , and  $\forall t \in T_{C_{com}}, \exists m = (t, tn) \in M_{loc}$ . These two last conditions ensure that the local and common domains share elements.

In order to estimate its bid on the sellable tasks, each bidder then builds a set of HTN planning problems corresponding to these tasks.

**Definition 8** (Bid Estimation Problems). Let  $\mathcal{D}_{bid}$  be the bid domain built by the bidder, and  $(\mathcal{H}_\delta, C_\delta, s_\delta, L_\delta)$  the received item for sale. Then, for each label  $l \in L_\delta$ , we define a bid estimation problem  $\mathcal{P}_l = (\mathcal{D}_{bid}, s_l, tn_l)$ , with:

- $s_l$  the initial state, containing state  $s_\delta$ , the associated constants  $C_\delta$ , and a bidder local initial state  $s_{loc}$ ;
- $tn_l$  the task network containing only the task in  $\mathcal{H}_\delta$  with label  $l$ .

The bidder then solves each bid estimation problem  $\mathcal{P}_l$ . In practice, we generate an HDDL model corresponding to  $\mathcal{D}_{bid}$  and  $\mathcal{P}_l$ , and call an HTN solver. If the solver returns a solution plan to  $\mathcal{P}_l$ , then the solution plan cost  $c_l^*$  is used as a bid value for  $l$ . Thereby, the bids from each bidder are defined as  $b = (L_{bid}, C_{bid}^*)$ , with  $L_{bid} \subseteq L_\delta$  set of labels on which the bidder bids and  $C_{bid}^*$  a set of costs associated to each label  $l \in L_{bid}$ .

### C. Winner determination

Once the auctioneer has received the bids from the bidders, it allocates tasks by solving the WDP. To do so, it builds a HTN problem  $\mathcal{P}_{wdp} = (\mathcal{D}_{wdp}, s_{wdp}, tn_{wdp})$ , where  $\mathcal{D}_{wdp}$  is the WDP domain that corresponds to the common domain  $\mathcal{D}_{com}$  extended with:

- one primitive task that corresponds to *allocating* a task to a robot. This task is modeled so that a robot can be allocated only one task at each round, and integrates a cost predicate corresponding to bids. Each task to sell in  $L_\delta$  can then be decomposed in this allocation task through a method,
- in case a task is not allocated, we need a second primitive task that corresponds to *reselling* the task in the next round. Each task to sell can also be decomposed into this reselling task,
- new symbols in the associated language  $\mathcal{L}_{wdp}$  that correspond to these new tasks, predicates and variables.

$s_{wdp}$  is the auction initial state  $s_\delta$  extended with the bids received from the bidders, and  $tn_{wdp}$  is the task network encoding the auction HTN  $\mathcal{H}_\delta$ .

Finding a task allocation then corresponds to solving  $\mathcal{P}_{wdp}$ , minimizing the total cost, i.e. the sum of bids on the allocated tasks. To give a sound problem to the HTN solver, we need to define a cost to the reselling primitive tasks (otherwise the solver would interpret a null cost, and reselling all tasks would then always be the best solution).

The value of the resale cost is actually a way to control the allocation scheme: we can indeed set the resale cost so that it allocates as much tasks as possible, or on the contrary, decide to keep a task for the next round instead of allocating it. We defined three domain-independent strategies to set the resale cost:

- *MaxB*: A pessimistic strategy about the bids that will be received in future rounds, and that should therefore allocate as many tasks as possible at once. For this strategy, the resale cost of each node is set as the maximum value of the received bids for this node plus one. If there is no bid on the node, the resale cost is set as the sum of the children resale costs plus one if this is an *AND* node, or the maximum of children resale costs plus one if this is an *OR* node (as in [19], we assume a bid is always made on leaf nodes).
- *MinB*: An optimistic strategy that may choose to not allocate a task when a better allocation is expected during a further round. For this strategy, the resale cost of each node is set as the minimum value of the received bids for this node plus one. If there is no bid on the node, the resale cost is set as the sum of the children resale costs plus one if this is an *AND* node, or the minimum of children resale costs plus one if this is an *OR* node.
- *MaxBL*: A mixed strategy being either optimistic or pessimistic, depending on the received bids. The resale cost is defined as with the *MaxB* strategy, without considering bids on non-leaf nodes, i.e. systematically using the costs of children.

### D. Auctioning on several rounds

Depending on the allocation and resale decisions of the auctioneer, auctioning on several rounds can sometimes be necessary. To begin a new round, the auctioneer creates a new item for sale  $\delta'$  by removing the tasks that have been allocated. Of course, each bidder considers the tasks it has been awarded during previous rounds to bid, by integrating its current allocated tasks in the bid estimation task networks  $tn_l$  (Def. 8). The auctioning cycle goes on until all tasks have been allocated.

## V. EVALUATION

### A. Experimental setup

In order to evaluate our approach, we consider a coverage mission of the naval minehunting context. As we laid down the foundations of our approach in this paper, this evaluation is done considering an abstract initial allocation problem. The experiments are performed in simulation using a symbolic representation of the world.

In this coverage mission we consider two kinds of primitive actions: area coverage and motion between two areas. Each action has a specific cost evaluation function, depending on the area surface and the distance to travel. The objective is to minimize the global cost that includes motion and covering actions for all robots. A series of auction HTNs are randomly generated for evaluation purposes: they differ in the number of tasks to allocate in the decomposition, and in the nature of considered ordering constraints. We generated 3 kinds of problems:

- *Unordered* problems, in which tasks are not constrained,

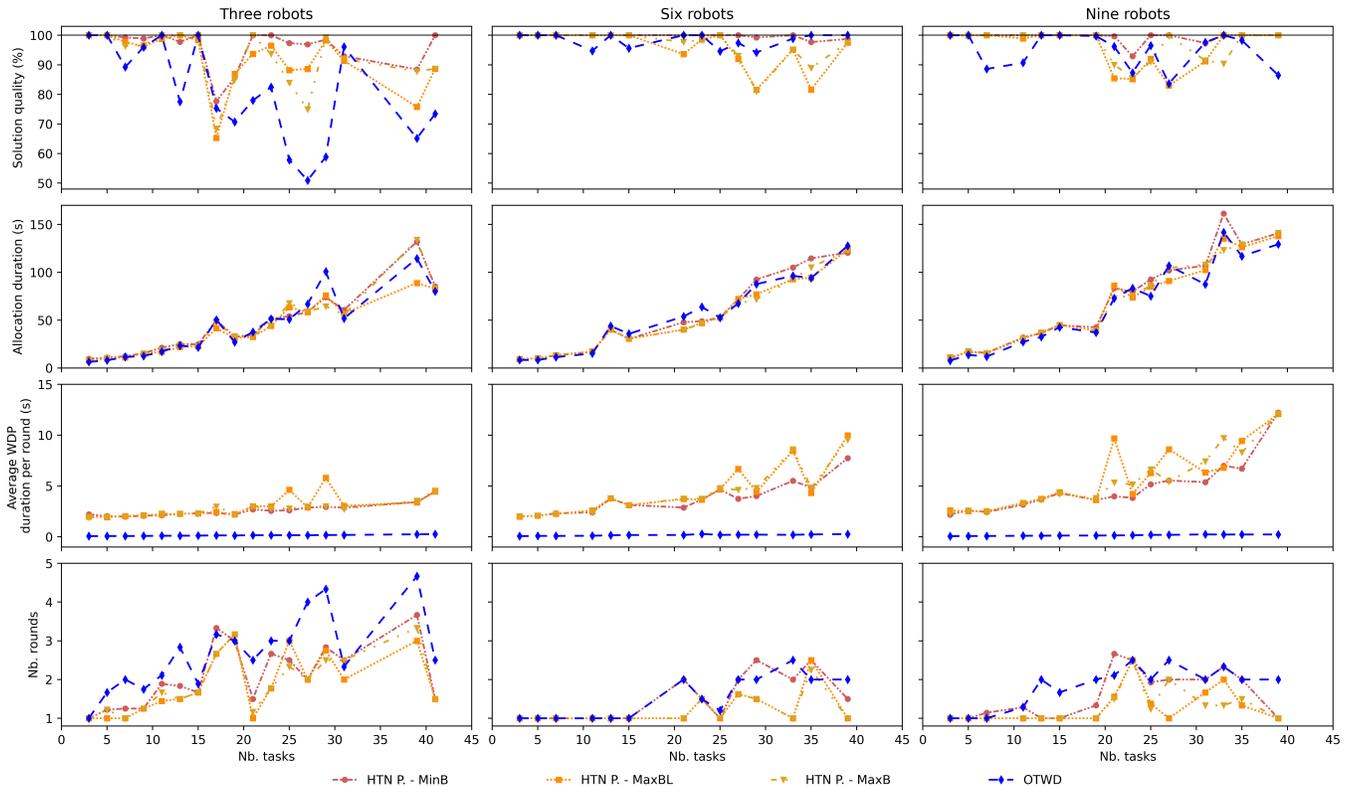


Fig. 3. Results for unordered HTNs showing the quality (top), allocation duration (2nd row), WDP duration (3rd row) and number of necessary rounds (bottom) for 3, 6 and 9 robots (respectively left, middle, and right columns). The three *MinB*, *MaxB*, and *MaxBL* approaches are compared to the centralized reference (continuous line) and the *OTWD* algorithm.

- *Totally-ordered* problems, in which decomposition of a task into children tasks enforces a total order on the children,
- *Partially-ordered* problems, in which some constraints are randomly generated.

To optimize the bid estimation and WDP solving, we use the PANDA solver [14]. We evaluate our approach considering the three *MaxBL*, *MaxB* and *MinB* strategies for the resale costs. The results are compared with the *OpTradWinnerDetermine* (*OTWD*) algorithm [19] – only for unordered problems, as *OTWD* does not handle ordering. In order to assess the optimality of the approaches, a reference solution is provided by solving the complete allocation at once in a centralized manner using PANDA. We generate HTNs with a varying number of tasks going from 3 to 40 and with 3, 6 and 9 robots. For each of these values, the results are averaged over 9 runs of different HTNs. The decision architecture is implemented using ROS2 and involves independent nodes working in parallel. The runs are realized on a PC with an intel® core™ i7-9750h CPU @ 2.60GHz and 16 GB.

### B. Results

Figure 3 presents the average of four metrics for the unordered HTNs, as a function of the number of tasks and robots: solution quality, allocation duration, WDP duration per auction round, and number of auction rounds.

Regarding the solution quality, our HTN planning approach with the *MinB* strategy for resale costs globally outperforms other approaches, including the *OTWD* algorithm: over all the tests, we perform as well as *OTWD* in 92% of the samples and outperform it in 55% of the samples. As expected, *MinB* always performs better than *MaxB* and *MaxBL*, by refining more the solution at the expense of more rounds. By being either optimistic or pessimistic on the next round, *MaxBL* has a more unpredictable performance.

Regarding the average WDP duration per round (third row), our approaches are always longer than *OTWD* (which is by its greedy nature almost instantaneous). But the required time increases only linearly with the number of tasks, and the differences are not significant for the total allocation duration that includes bids estimations (second row).

The number of rounds decreases with the number of robots. For 3 robots, the number of rounds is 2.5 on average for *OTWD*, and 1.9, 1.8 and 1.7 for the HTN planning with *MinB*, *MaxB*, and *MaxBL* strategies respectively. Since the communication bandwidth is directly bound to the number of rounds and robots, our approach is better suited than *OTWD* to communication restricted environments.

Finally, Fig. 4 shows the solution quality on the HTNs with ordering constraints. As *OTWD* cannot integrate such constraints, we only compared with the global reference. These results demonstrate the capability of our approach to handle ordering constraints while solving allocation problems

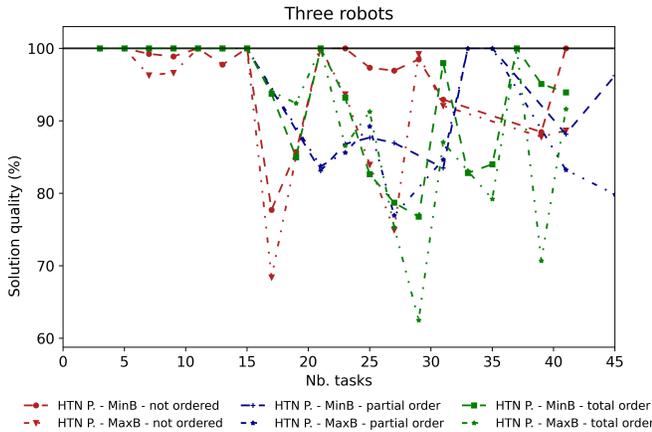


Fig. 4. Solution quality on unordered, partially-ordered and totally-ordered HTNs for 3 robots, in comparison to the centralized reference.

with good performances, without any noticeable additional computation time than for unordered problems.

## VI. CONCLUSION

We proposed and formalized a decentralized approach to handle the MRTA problem which relies on an auction-based allocation scheme combined with HTN planning. The approach reasons on hierarchically linked tasks to interleave allocation and decomposition. The approach experimentally demonstrated its ability to produce allocations with a better quality than the reference state of the art algorithm. We proposed and evaluated different *resale cost strategies* to control the allocation process: they allow to either optimize the quality of the allocations or to reduce the necessary communications.

Up to now, the approach has only been evaluated on initial complex task allocation problems. Nevertheless, it can intrinsically be used for mission execution and supervision, to cope with execution errors and with the appearance of new tasks, thus being able to tackle online plan reparation problems. A significant improvement will consist in explicitly model and handle partially-ordered problems combined with causal constraints – preliminary work extending our framework to that end are proposed in [26].

Finally, in order to have a finer control, future work will also explore alternate methods to set the resale cost, using generic HTN heuristics or mission dependent designs.

## ACKNOWLEDGEMENTS

We would like to express our gratitude to the PANDA developers for their accessibility, and especially to Gregor Behnke for the help provided.

## REFERENCES

- [1] B. Gerkey and M. Matarić, “A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems,” *IJRR*, 2004.
- [2] A. Khamis, A. Hussein, and A. Elmogy, “Multi-robot Task Allocation: A Review of the State-of-the-Art,” *Cooperative Robots and Sensor Networks*, 2015.
- [3] A. Nègre, H. Zhang, O. Marceau, B. De Saporta, D. Laneville, and F. Dufour, “Stochastic control for underwater optimal trajectories,” in *IEEE Aerospace Conference*, 2012.

- [4] O. Marceau and J.-M. Vanpeperstraete, “AUV optimal path for leak detection,” in *OCEANS*, 2017.
- [5] E. Chauveau, C. Lesire, and F. Chaillan, “Integration of Autonomous Heterogeneous Systems for Decision Making Autonomy in Naval Defence: A Position Paper,” in *OCEANS*, 2019.
- [6] E. Artusi and F. Chaillan, “Automatic recognition of underwater acoustic signature for naval applications,” in *Maritime Situational Awareness Workshop*, 2019.
- [7] Q. Oliveau and H. Sahbi, “Learning Attribute Representations for Remote Sensing Ship Category Classification,” *Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2017.
- [8] G. Ferri, A. Munafò, A. Tesei, P. Braca, F. Meyer, K. Pelekanakis, R. Petrocchia, J. Alves, C. Strode, and K. LePage, “Cooperative robotic networks for underwater surveillance: an overview,” *IET Radar, Sonar & Navigation*, 2017.
- [9] M. Dias, R. Zlot, N. Kalra, and A. Stentz, “Market-Based Multirobot Coordination: A Survey and Analysis,” *Proceedings of the IEEE*, 2006.
- [10] P. Bercher, R. Alford, and D. Höller, “A survey on hierarchical planning - one abstract idea, many concrete realizations,” in *IJCAI*, 2019.
- [11] G. Ferri, A. Munafò, A. Tesei, and K. LePage, “A market-based task allocation framework for autonomous underwater surveillance networks,” in *OCEANS*, 2017.
- [12] M. Otte, M. J. Kuhlman, and D. Sofge, “Auctions for multi-robot task allocation in communication limited environments,” *Autonomous Robots*, 2019.
- [13] P. Bechon, C. Lesire, and M. Barbier, “Hybrid planning and distributed iterative repair for multi-robot missions with communication losses,” *Autonomous Robots*, 2020.
- [14] P. Bercher, S. Keen, and S. Biundo, “Hybrid Planning Heuristics Based on Task Decomposition Graphs,” in *SOCS*, 2014.
- [15] N. Kalra, D. Ferguson, and A. Stentz, “Hoplités: A Market-Based Framework for Planned Tight Coordination in Multirobot Teams,” in *ICRA*, 2005.
- [16] M. B. Dias, B. Ghanem, and A. Stentz, “Improving cost estimation in market-based coordination of a distributed sensing task,” in *IROS*, 2005.
- [17] S. Botelho and R. Alami, “M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement,” in *ICRA*, 1999.
- [18] E. Nunes and M. Gini, “Multi-robot auctions for allocation of tasks with temporal constraints,” in *AAAI*, 2015.
- [19] R. Zlot and A. Stentz, “Market-based Multirobot Coordination for Complex Tasks,” *IJRR*, 2006.
- [20] Y. Liu, J. Yang, Y. Zheng, Z. Wu, and M. Yao, “Multi-robot coordination in complex environment with task and communication constraints,” *International Journal of Advanced Robotic Systems*, 2013.
- [21] A. M. Khamis, A. M. Elmogy, and F. O. Karray, “Complex task allocation in mobile surveillance systems,” *JINT*, 2011.
- [22] S. Koenig, P. Keskinocak, and C. Tovey, “Progress on Agent Coordination with Cooperative Auctions,” in *AAAI*, 2010.
- [23] K. Erol, J. Hendler, and D. S. Nau, “HTN planning: Complexity and expressivity,” in *AAAI*, 1994.
- [24] D. Höller, G. Behnke, P. Bercher, S. Biundo, H. Fiorino, D. Pellier, and R. Alford, “HDDL: An Extension to PDDL for Expressing Hierarchical Planning Problems,” in *AAAI*, 2020.
- [25] D. Nau, Y. Cao, A. Lotem, and H. Munoz-Avila, “SHOP: Simple hierarchical ordered planner,” in *IJCAI*, 1999.
- [26] A. Milot, E. Chauveau, S. Lacroix, and C. Lesire, “Solving Hierarchical Auctions with HTN Planning,” in *4th ICAPS workshop on Hierarchical Planning (HPlan)*, 2021. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-03289854v2>