A Task Decomposition using (HDec-POSMDPs) Approach for Multi-robot Exploration and Fire Searching

Ayman Elshenawy Elsefy

Al-Azhar University

Abstract— In this paper, hierarchical control architecture for coordinated multi-robot systems (MRS) task decomposition is presented; based on a hybrid decentralized Partially Observable Semi-Markov Decision Processes (HDec-POSMDPs). In this architecture, robots can make their own decisions according to their locally collected data with limited communication between a robot team. In this proposed architecture, the global task is decomposed into multiple local sub-tasks using divide and conquer design, each task is described as a set of regular languages. MRS are modeled as a discrete event system and each robot is represented by a deterministic finite state automaton model. Direct Cross-Entropy (DICE) can be used for searching the space of the best frontier cells to solve the Dec-POSMDP and each sub-task is assigned to one or more robots to be executed. The proposed algorithm is implemented, tested and evaluated in the computer simulator. By using this architecture, the task execution time is minimized, the fire sources cluttered in an environment have been searched in an effective manner and the performance of MRS has been enhanced with respect to energy consumption and communication load; when they are used for exploring different environments as well as when they are used for detecting the sources of the fire and reporting about them.

Keywords— task assignment, task decomposition, multi-robot systems, HDec-POSMDPs, finite state automata, discrete event system, exploration and fire searching, cooperative multi-robot.

DOI: 10.21535/ijrm.v7i1.1026

I. INTRODUCTION

THE exploration and search about a special object in an unknown environment are sometimes extremely dangerous processes to be performed by a human. Multi-Robot Systems (MRS) can be used to perform such processes by covering the whole environment in a timely manner and detecting the required object[1][2]. There are many issues that affect the performance of MRS, such as exploring the same region by more than one robot, assigning more than one robot to cover the same region, robots collision with each other when they are crossing a path, etc.[3]. To improve the performance of such processes in complex environments, MRS are used to accomplish a task in an effective way and in a shorter time; since they use a parallel and simultaneous execution. For problems that involve a set of distributed tasks, MRS assign a set of tasks to a set of robots and collect the behavior of these tasks to constitute the global task[2][4][5].

MRS are classified into cooperative robots that have different sensory, sizes, shapes, function capabilities, and swarm robots that are identical, larger in number and small in size[6]-[9]. MRS exploration team performs a sequence of actions to achieve a specific mission; which is known as mission planning. Mission planning is usually divided into two major modules (i) task planning and (ii) motion planning. Task planning includes task decomposition and task assignment. Task decomposition is defined as the process of dividing a global complex task into several sub-tasks that can be performed independently by a set of robots. It depends on the nature of decomposition process that is observed in a lot of applications and activities involving foraging, hunting, nest excavation, power grids, firefighting, transportation networks and garbage disposal[10][11].

The application of the task-decomposition functions improves the performance of MRS exploration because of the following reasons (i) it decreases the effect of interference and competition for shared resources by applying physical separation of MRS team, (ii) each sub-task can be assigned to the suitable robot, (iii) Increases the efficiency of performing a task due to energy conservation for each robot, lower data process, and high fault tolerance, (iv) Reduces the explored space by each robot in the team and improves the quality of the map in the exploration process[10].

Robots can perform several types of tasks: (i) elemental or atomic tasks that involve a single un-decomposable action to be performed by a single robot, (ii) simple compound task, is a decomposable simple task that may be assigned and performed by different robots or by the same robot, (iii) compound task that can be decomposed into a set of simple compound sub-tasks since it provides a single fixed way of decomposing the task into sub-tasks, (v) complex tasks, are ones for which there are multiple possible ways for decomposing the task and which can be assigned to multiple robots. Each sub-task in a decomposition of complex tasks may be simple, compound, or complex[12][13]. As shown in Figure 1, the dashed circles indicate the assignment of tasks to robots. Shaded circles represent elemental tasks, while shaded rectangles represent decomposable tasks, and their decomposition into elemental tasks is shown by a tree-like structure. The overlapping (nested) trees in the rightmost figure illustrate several ways to analyze and decompose the complex task.

Corresponding author:
Ayman Elshenawy Elsefy (e-mail: eaymanelshenawy@azhar.edu.eg)
This paper was submitted on August 13, 2018, and accepted on August 24, 2019.

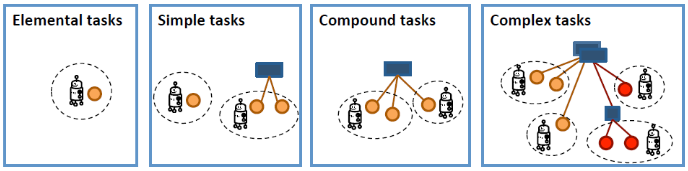

Figure 1 Variety type of tasks

Task In order for MRS to execute a global task on the basis of the task decomposition, the following policies must be available (i) local control policy to control the sub-tasks which are performed by each individual robot, and (ii) interaction control policy to handle the interaction between each robot and the other team members. The design of individual local control policies and interactional control policies is one of the most challenging problems in MRS to execute the global task[14][15]. The decomposition of the global task of MRS can be implemented using two hierarchical control architectures, the bottom-up architecture and the top-down architecture.

In the bottom-up design architecture, the local control policies and the interactional control policies are predefined and gained from natural behaviors. The global task of MRS is formed from simple individual local control interaction rules among robots[14][16]. It is a simple approach, but it does not achieve high-level specifications of the task when the global task changes. It is inefficient and time-consuming. A lot of researchers try to introduce a bottom-up design architecture based on Linear Temporal Logic (LTL)[17], automata-based receding horizon technique[20], and a graph-search (Bellman-Ford) method of finite (LTL)[20]. These techniques guarantee some enhancement represented in scalability, flexibility, and robot failure detection[17], enhancement of the communication description, commands, and control structure between the networked robots[18]. But they do not ensure fairness when each robot has a local plan as an infinite sequence of actions[17], do not support a distributed negotiation between two groups[18].

The top-down design architecture is more efficient than the bottom-up approach. It explicitly partitions the global task into individual local sub-tasks. Each sub-task is executed by an individual robot, and the global task is achieved from this local information[14][16][17]. A lot of researches depended on top-down design architecture using a Local Voronoi Decomposition (LVD) algorithm[3], finite automata, LTL and modified L star algorithm[3][14][16]. They prove that the top-down design architecture dynamically changes the exploration map, can add or remove robots during the exploration without affecting the task execution[3] and enhances the system robustness by utilizing redundant functions of robots[14].

All the above researches present a solution for MRS task decomposition problem for simple global tasks, and the results of these architectures do not handle robustness to small perturbations[19], the generated paths are not efficient enough to produce a high-quality map for the exploration process[3], they are proved theoretically but it does not implement in real robots[22].

In this paper, a top-down design control architecture is proposed for solving the task decomposition problem of cooperative and coordinated MRS to perform the exploration of an unknown environment and search for the cluttered fire source in this environment. A hybrid decentralized Partially Observable Semi-Markov Decision Processes (HDec-POSMDPs) control theory is proposed to design hierarchical control architecture for robotic task assignment approach in which robots can make their own decisions according to their local information with limited communication between a robot team. In this architecture, MRS are modeled as a Discrete Event System (DES) and each robot is represented by a Deterministic Finite State Automata (DFSA) model to express a large class of tasks[9][16][21]. The decomposition is done in such way that all individual tasks are done by individual robots and the global task requirements are guaranteed by control design approach.

The paper is organized as follows: in Section II, the system overview and formulation of the problem for a task decomposition of MRS is discussed, Section III discusses the HDec-POSMDP approach for Multi-robot Exploration and Fire Searching Tasks are addressed. To illustrate the task decomposition, an implementation result for a coordinated MRS is produced in Section IV. Finally, our work is concluded with a suggestion for future works that are presented in section V.

II. System Overview and Problem Formulation

The global task of exploration and fire searching process are described as a DFSA that includes a set of all possible events, local control policies and the interactive control policies. A Dec-POMDP is used as a decision-making process for synchronization problems that operate under uncertainty; depending on a set of observations collected from local robots. Each individual robot selects in parallel an action that translates an immediate reward for a state, action and observation space based on Markovian models and local observations. For a cooperative MRS, each individual robot shares its reward function with other robots team in a decentralized manner; based on the action of all robots.

A. Robotic Coordination

A continuous HDec-POMDP can be easily converted to discrete HDec-POSMDP in which actions are chosen in semi-Markov setting; using actions with the probability of completion. The HDec-POSMDP framework can be formulated as follows:

Given (1) $R_i = \{1,2,\ldots\ldots n\}$ is a set of finite robots, (2) $\mathrm{Room}_j = \{1,2,\ldots\ldots m\}$ is a set of rooms, (3) $\dot{\check{S}}$ is a set of states that contains robots and environmental states; it can be represented as $\dot{\check{S}} = \dot{\check{S}}^i \times \dot{\check{S}}^e$ where $\dot{\check{S}}^i$ is the state space of ith robots and $\dot{\check{S}}^e$ is the environmental state. (4) $\acute{\breve{U}}_i$ is a set of robot actions $\acute{\breve{U}}_i = \{u_1, u_2, \ldots\ldots, u_n\}$. (5) $\acute{\breve{O}}$ is a set of observations obtained by robots and environmental observations; it can be represented as $\acute{\breve{O}} = \acute{\breve{O}}^i \times \acute{\breve{O}}^e$ where $\acute{\breve{O}}^i$ a set of observations obtained by ith robot and $\acute{\mathrm{O}}^{e}$ is the environmental observations which are functions of environmental state $\mathring{\check{\mathrm{S}}}^{e}$. (6) $\check{\mathrm{R}}$ is a reward function $\check{\mathrm{R}}: \mathring{\check{\mathrm{S}}} \times \acute{\mathrm{U}}_{i} \rightarrow \acute{\mathrm{R}}$ which is the immediate reward function when a robot beings in state $\check{\mathrm{S}}$ and takes an action $\acute{\mathrm{U}}_{i}$. In a HDec-POMDP, robots can make decision based on their action-observation histories as follows[23]:

$$H_i^t = \{o_i^0, u_i^0, \dots \dots, o_i^{t-1} u_i^{t-1}, o_i^t\}$$

The solution of HDec-POMDP is a collection of decentralized policies $\dot{\rho} = \{\acute{\rho}^1, \acute{\rho}^2, \dots \dots, \acute{\rho}^n\}$. In general, each robot does not have an access to the observations of other robots; so each policy depends only on the local information of each robot. As a result, $\acute{\rho}^i$ maps the individual full action-observation history of the ith robot to its next action: $u_i^t = \acute{\rho}^i \times \mathrm{H}_i$. The value related to decentralized policy starting from state distribution $\mathrm{Ƥ} = p(\mathring{\check{\mathrm{s}}})$.

$$V^{\acute{\rho}}(\mathrm{Ƥ}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \Upsilon^t \, \check{\mathrm{R}}(\mathring{\check{\mathrm{s}}}_t, \acute{\mathrm{u}}_t) | \acute{\rho}, p(\mathring{\check{\mathrm{s}}}_0) = \mathrm{Ƥ}\right] \quad \textbf{(1)}$$

The solution to HDec-POMDP is to find the optimal policy with the highest value starting at the initial state distribution:

$$\acute{\rho}^* = \mathrm{argmax}_{\acute{\rho}} V^{\acute{\rho}} \quad \textbf{(2)}$$

where $\acute{\rho}^*$ is the action that each robot must take based on its action history and the received observations. The result is a sequential decision-making process that ultimately leads to actions and observations.

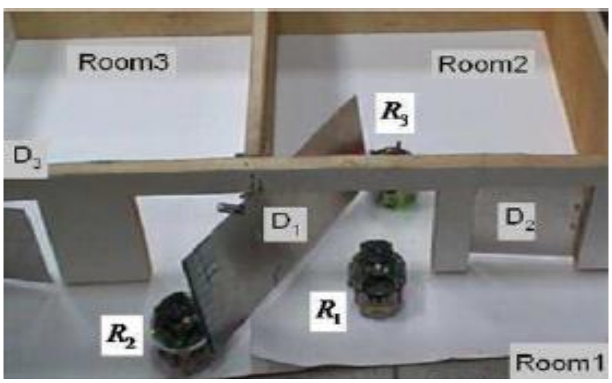

Figure 2 The environment of MRS coordination for exploration and fire searching

To demonstrate the proposed scenario as depicted in Figure 2, a cooperative MRS consists of four heterogeneous robots of the same size and different capabilities R_i, $i = \{1,2,3,4\}$, are employed to explore and search for the fire sources in an environment of four Rooms $\mathrm{Room}_j, j = \{1,2,3,4\}$ of different structures. Initially, all MRS members are located in $Room_1$, they have no knowledge about the environment. The robots R_1 and R_2 are equipped with a fire detector sensor capabilities, while R_3 and R_4 are equipped with pushing capabilities[3][9][16][21]. All doors are equipped with spring to close automatically since there are no forces to keep them open. $Room_2$ is accessible from $Room_1$ by two-way door D_1 and one-way door D_2 and, $Room_3$ is accessible from $Room_1$ by two two-way doors D_1 and D_3, and $Room_4$ is accessible from $Room_1$ by one-way door D_4 and two-way door D_3, a portion of this environment is shown in Figure 2.

There are some basic assumptions about the environment such as (i) Flexibility in task assignment: one or more robots can be assigned to any Room. (ii) Flexibility in task control: if the assigned task is not completed successfully due to communication failure or communication loss with other robots, the control design methodology is capable of reassigning this task to another functional robot of MRS. (iii) Coordination of MRS: the incapable robot can request a help from other capable robots to perform a specific task and the other can respond to this help[9][20][16][22].

B. Automata Models of Multi-robot Systems

Consider The Considering MRS that consist of four robots R_i, $i = \{1,2,3,4\}$, the behavior of MRS is formalized as discrete-event system (DES), and each robot is represented by a deterministic finite state automaton (DFSA) model $G_i = (\mathring{\check{\mathrm{S}}}, \acute{\mathrm{U}}_i, \mathring{\check{\mathrm{s}}}_0, \delta_i)$, where $\mathring{\check{\mathrm{S}}}$ is a sequence of local states, $\acute{\mathrm{U}}_i$ is a sequence of local events (a set of actions that can be performed by an individual robot), $\delta_i: \mathring{\check{\mathrm{S}}} \times \acute{\mathrm{U}}_i \rightarrow \mathring{\check{\mathrm{S}}}$ is the state transition function, and $\mathring{\check{\mathrm{s}}}_{0,i} \in \mathring{\check{\mathrm{S}}}$ is the local initial state. δ_i can be extended to $\mathring{\check{\mathrm{S}}} \times \acute{\mathrm{U}}_i^{\,*} \rightarrow \mathring{\check{\mathrm{S}}}$ in a natural way. The global task extended the (DFSA) G_i of each robot to generate a regular language that is accepted by a sequence of events and is given by: [9][16][21]

$$L(G_i) = \{u \in \acute{\mathrm{U}}_i^{\,*} | \delta(\mathring{\check{\mathrm{s}}}_{0,i}, u) \ is \ defined\}$$

The global logical behavior of MRS is modeled as the collection of concurrent operations of each individual robot and is given by $\mathrm{G} = ||_{i=1}^{4} G_i$, where || means the parallel composition of DFSA [9], [25], [26]. Define $\acute{\mathrm{U}} = \cup_{i=1}^{4} \acute{\mathrm{U}}_i$ as the global event set which consists of controllable event sets $\acute{\mathrm{U}}_c = \cup_{i=1}^{4} \acute{\mathrm{U}}_{i,c}$ and uncontrollable $\acute{\mathrm{U}}_{uc}$ event sets, therefore, $\acute{\mathrm{U}} = \acute{\mathrm{U}}_c + \acute{\mathrm{U}}_{uc}$ and $P_i: \acute{\mathrm{U}}^* \rightarrow \acute{\mathrm{U}}_i^{\,*}$, as the natural projection for robot G_i with the inverse projection that is denoted by P_i^{-1}. The collective behavior of MRS is captured by the synchronous or parallel product of $L(G_i)$: $L(G) = ||_{i=1}^{4} G_i = \cap_{i \in N} P_i^{-1}(L(G_i))$. Assuming that in MRS scenario any events shared by more than one robot also agree on the status of controllability[9][15][16][21]

$$\acute{\mathrm{U}}_{i,c} = \acute{\mathrm{U}}_c \cap \acute{\mathrm{U}}_i$$

During MRS coordination task, there are two different cases of the robot failures and robot repairs which are considered in DFSA model design. A robot failure may be (i) Temporary failure: a failure occurs and the robot will be repaired to accomplish its task, and (ii) Permanent failure: a failure occurs and the robot is not able to accomplish its task, its task must be reassigned to another robot[9].

The global task requirements of DFSA models are synthesized using individual task completion requirement models. There is a variety of models used for task completion, such as: (i) the task is performed by only one Robot, (ii) the task is performed by one robot or by the other, and (iii) the task is performed first by one robot and subsequently by another one.

In this paper, the exploration and fire searching problems are modeled based on a second and third model[27].

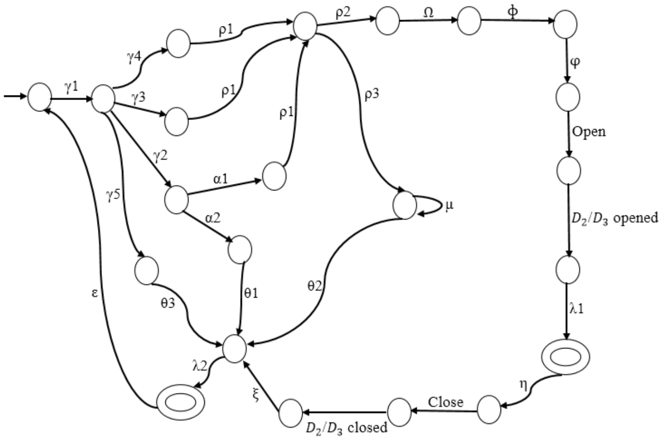

Figure 3 Desired behavior task of R_i

The DFSA can be considered a controller for each robot i that takes a set of robot observations Ő as inputs and outputs and actions Ű for the robot to execute at each step. Once the other parameters (such as initial state, state transition, and output function) have been specified for each robot, a set of controllers is defined as shown in Figure 3, which chooses actions based on the current node and transition based on the observations that have been seen.

For a DFSA $\mathcal{F}_i$and for robot R_i, which induces a policy ṗ, the DFSA generates mappings from histories to actions that are, execution proceeds with $Ű_i$ followed by a transition in the controller $Ṩ = \delta^i(ṩ_{0,i}, ő^{ei}) \in Ṩ$, an action selected at the next cell, and so on. Note that although the domain is stochastic, the policy is deterministic. The policy can proceed for an infinite number of steps due to the loops in the controller. A set of such DFSAs (one for each robot) can be defined and parameterized by joint output functions $Ű_i = \{u_1, u_2, \dots \dots, u_n\}$ and joint transition (input) functions $\delta^i(ṩ_i, ő^{ei}) = \{\delta^1(ṩ_1, ő^{e1}), \dots, \dots, \dots, \delta^n(ṩ_n, ő^{en})\}$. The DFSAs can be evaluated at a given initial joint state distribution Ƥ and initial joint node ṩ over interval k as follows:[23]

$$V^{ṗ}(Ƥ, ṩ, ṩ^e) = Ř(Ƥ, ṩ^e, Ű_i; ṗ) + \sum_{\boldsymbol{t}_{k,ő^e}} \Upsilon^t p(ő^e | Ƥ, ṩ^e, Ű_i; ṗ) \quad \textbf{(3)}$$

$$p(\boldsymbol{t}_k | Ƥ, ṩ^e, Ű_i, ő^e; ṗ) V^{ṗ}(Ƥ_{ő^e, Ű_i}, {ṩ^e}_{ő^e, Ű_i}, \delta^i(ṩ_i, ő^{ei}))$$

Therefore, the value of the policy starting from the initial cell for each robot can be calculated as $V^{ṗ} = (Ƥ_0, {ṩ^e}_0, ṩ_0)$.

In order to describe the collective behaviors for robot R_i, the actions of R_i can be fired by the roles as shown in Figure 3, and result in uncontrollable actions. Small controllable events are taken into consideration to handle the occurrence of uncontrollable events operation. For example, the measuring or sensing surrounding action (γ1) is controllable, *i.e.*, it might be allowed or not by the roles. However, the detecting results are uncontrollable, *i.e.*, the results may be one of the following, frontier cells are detected (γ2), fire source is detected (γ3), no frontier cells or fire source are detected (γ4), and obstacles are detected (γ5). In other words, these result events are allowed to occur. Also, there are uncontrollable events such as no fire source found (α2) and for communications success (ρ2) or communications fail (ρ3) between two robots on the environment and the communication infrastructure on two corresponding robots, all these actions are not upon the role[22]. A set of probable actions are shown in Table 1.

Table 1 A set of possible actions

Event	Explanation
Ω	Robot R_i receives the service request, $i = 1, 2, 3, 4$.
ϕ	Robot R_i approaches the door D_2 or D_3, $i = 3, 4$.
Φ	Robot R_i localizes itself at the door D_2 or D_3, $i = 3, 4$.
η	Robot R_i heads for Room4, $i = 1, 2, 3, 4$.
ξ	Robot R_i stays at Room4 for next request, $i = 1, 2, 3, 4$.
Open	Command for moving forward to open D_2 or D_3.
Close	Command for moving backward to close D_2 or D_3.
D_2/D_3 open	The status that D_2 or D_3 is opened by R_3 and R_4.
D_2/D_3 closed	The status that D_2 *or* D_3 is closed by R_3 and R_4.
γ1	Measured surroundings controllable
γ2	Frontier cells are detected uncontrollable
γ3	Fire source are detected uncontrollable
γ4	No frontier cells or fire source are detected uncontrollable
γ5	Obstacles are detected uncontrollable
ρ1	Establish a link with other controllable
ρ2	Link establishment succeeded uncontrollable
ρ3	Link establishment failed uncontrollable
θ1	Select a final target under γ2 controllable
θ2	Select a final target under γ4 and ρ3 controllable
θ3	Avoidance collision controllable
ε	Update map controllable
μ	Quit the communication link controllable
λ1	Move to a first target controllable
λ2	Move to a final target controllable
α1	Found fire source controllable
α2	No fire source is found uncontrollable

The coordinated events are described as follows: ε is the state update event, *i.e.*, after arriving the target cell the robot R_i updates the target cell as the new current state, θ1 is an action that R_i selects a final target according to Direct Cross-Entropy method; which is discussed in the next section. In a similar way, θ2 is the action that R_i selects a final target based on limited available information (no frontier cells, no fire sources are detected and communication between the robots failed). It selects a final target according to maximum utility. In this case the (avoidance of the collision), it means selecting their target from the unfinished job list, θ1 and θ2 both are during independent exploration and fire search period of multi-robot coordination.

In the period in which Multi-robot team does not need cooperation during the exploration and fire searching processes, the collective behavior of R_i is $(\gamma 1\ \gamma 2\ \alpha 2\ \theta 1\ \lambda 2\ \varepsilon)^*$. In the period that needs cooperation and coordination among robots and for the cases of communication success, the collective behavior consists of three strings: $(\gamma 1\ \gamma 4 \rho 1 \rho 3 \theta 2\ \lambda 2\ \varepsilon)^*$ or $(\gamma 1\ \gamma 3 \rho 1 \rho 3 \theta 2\ \lambda 2\ \varepsilon)^*$ or $(\gamma 1\ \gamma 2 \alpha 1 \rho 1 \rho 3 \theta 2\ \lambda 2\ \varepsilon)^*$. In the same period but in case of communication failure, the collective behavior consists of three strings:

$(\gamma 1\ \gamma 4 \rho 1 \rho 2 \Omega \phi \varphi \mathrm{Open} D_2/D_3\ \mathrm{open} \lambda 1 \eta \mathrm{Close} D_2/D_3\ \mathrm{closed} \xi \lambda 2 \varepsilon)^*$ or

$(\gamma 1\ \gamma 3 \rho 1 \rho 2 \Omega \phi \varphi \mathrm{Open} D_2/D_3\ \mathrm{open} \lambda 1 \eta \mathrm{Close} D_2/D_3\ \mathrm{closed} \xi \lambda 2 \varepsilon)^*$ or

$(\gamma 1\ \gamma 2 \alpha 1 \rho 1 \rho 2 \Omega \phi \varphi \mathrm{Open} D_2/D_3\ \mathrm{open} \lambda 1 \eta \mathrm{Close} D_2/D_3\ \mathrm{closed} \xi \lambda 2 \varepsilon)^*$.

Finally, the behavior of collision avoidance with obstacles or with other robots consists of only one string: $(\gamma 1 \gamma 5 \theta 3 \lambda 2 \varepsilon)^*$ which has the highest priority [22].

III. The Hybrid Decentralized Coordination Approach for Multi-Robot Exploration and Fire Searching

This proposed HDec-POSMDP framework for exploration and fire search consists of five stages as the following:

- **Initialization (Robot Deployment, selection of next location and Navigation Planning):** (i) the deployment strategy of a robot determines the startup locations for each robot in the environmental map. At the beginning of exploration process, robots are deployed randomly in the environment with restrictions that the distance between them is less than the scope of the communication range. (ii) Mobile robot navigation, the navigation of a robot includes three basic issues (a) building a map: by using a set of information sensor acquired by a robot, (b) localizing itself in this map: the exact position and orientation of a robot in the environment need to be determined at all times, and (c) planning a path: a robot should be able to generate a collision-free trajectory from its current pose to a desired target pose, the path is planned using EA* algorithm [1].
- **Collision avoidance:** During the navigation process, robots must avoid collisions with obstacles and other robots by implementing some rules between the robot team to make the navigation process more efficient as shown in algorithm1. The robot tries to detect a frontier or fire source cell during the navigation process. If the robot detects a cell which is a frontier or a fire source cell, it will go to the next step, if not it will end exploration process.
- **Target location determination and assignment:** As soon as a robot has determined its own role to move, it has to find another (next) location for further exploration of the environment. The next location is determined so that the following requirements are met: (1) allows to discover new and more unexplored areas, (2) less energy consumption, so that the robot must be very close to it to reach quickly, (3) Maintains network connectivity and does not break it, this paper is based on [1], and (4) does not induce overlaps between robot's sensor ranges. The candidate frontier cell is determined and assigned based on (Dec-POSMDPs).

Dec-POSMDPs is considered as a combinatorial optimization problem, where decision variables are the best frontier cells of robots. It provides solutions for computationally intractable, continuous and large space domains in terms of state, actions, and observation space with long horizons[23]. The divide and conquer strategy is used. In the divide part, each robot decides on the nearest cell, if it is inside its region or not; based on a utility function that is calculated depending on the position of all robots from this. In a conquer part, the winner robot will move to the target frontier according to the cell-based variant called Direct Cross-Entropy (DICE) that is used for searching the space of the best frontier cells in order to solve the Dec-POSMDP. The randomized sampling approach is used by Cross-Entropy (CE) method to solve the optimization problem of finding an optimal frontier cell as in Eq. (2).

Algorithm 1. Collision Avoidance

```
Measure a surrounding area around the robot.
if A ROBOT is in Region OR Obstacle Exists, then
  if a visited obstacle is detected OR probable collision OR
  non-visited obstacle, then
    Move to the nearest occlusion point which has not
    been visited.
  endif
elseif all obstacle cells are already visited, then
  Move to the least recently visited obstacle cells.
  if there is a possibility for interference, then
    if the robots in a direct collision path, then
      Reassign their targets.
    elseif one of frontiers is selected by more than one
    robot, then
      Give priority to the one which has higher utility.
    else
      Give priority to the one that is exploring a frontier.
    endif
  elseif No obstacle exists, then
    Move to the nearest frontier which does not coincide
    with any obstacle.
  else
    Continue exploration algorithm.
  endif
endif
```

$$f^* = \mathrm{argmax}_f\, v(f) \quad \textbf{(4)}$$

CE establishes a distribution state space of samples Ʒ (f; Γ), parameterized by Γ, that is used for sampling solutions f in the overall space of search. This procedure is summarized in Algorithm 2[23][24]. The process of fire searching is done in parallel with the frontier detection based on sensing the heat levels of objects allocated in the environment using the temperature sensors. Once the fire source is detected, the map is updated and the robot moves to its goal. If no fire source is detected, the robot will complete the exploration task to reach to winner frontier.

- **Path cost computation:** in this phase the path cost from the current robot to the best cell is computed using the steps listed in Algorithm 3. This step insures that a robot will be assigned to a goal candidate in a way that minimizes the total cost of the consequence movements. Multiple Traveling Salesman Problem (MTSP) method is used to assign the consequence of goals to a set of robots by ordering the goals and selecting the first not assigned goal to the appropriate robot[28]. A cluster-first, route-second heuristic approach is used for providing an approximate solution which is used for solving our problem which is based on an assignment of m clusters of the goals to robots. This can be summarized as the following:

First, a set of k clusters $\mathrm{C} = \{C_1, C_2, \dots \dots, C_k\}$ is determined based on the Euclidean distance between the selected frontier cells and the robot positions to provide clusters for which real paths to the goals are significantly longer than the expected ones [29]. Second, the TSP cost for the pair C_i, R_i is determined. Third, the first goal of the TSP tour from each non-empty cluster C_i is assigned to the robot r_i . Finally, the goal assignment is fixed if there is an empty set C_i.

Algorithm 2. Direct Cross-Entropy

- **input**: set of actions space, environmental observation space, learning rate $\tau \in [0, 1]$, number of robots R_i.
- **output**: best frontier policy f^*
  - Samples set is generated F from $\mathfrak{Z}$ (f;Γ_h), where h is the number of iteration and Γ_0 is the initial parameter vector.

    for $i = 1$ to R_i, **do**

    $\Gamma_{h+1} \leftarrow$ Maximum Likelihood of Γ using f^*

    $$\Gamma_{h+1} = \mathrm{argmax}_{\Gamma} \mathfrak{Z}\ (f^*;\ \Gamma)$$
  - Apply the smoothed update to

    $$\Gamma_{h+1} \leftarrow \tau\Gamma_{h+1} + (1-\tau)\Gamma_h$$
  - Repeat until convergence, return best sample f^*.

Algorithm 3. Compute path cost

- **Calculate** the distance between robots and frontier targets.

  $$L = \min(\{|P_1|, |P_2|, |P_3|, \dots \dots \dots \dots, |P_m|\})$$
- **Find** the lowest distance and assign the corresponding frontier to the winner robot.

  $$K = \min\{l, N_{max}\}, means$$
- **Decrease** the cost if the robot is already in a frontier target.
- **For each** robots R_i, **do**

  Move (r_i, P)

- **Global Map Construction, Broadcasting and Decision to Move:** When the next target location is reached by a robot, new information is gathered, added to the current information, updated map information and a global map is created. This new information is broadcasted to other robot team. When a robot receives the message format of broadcasted information, it compares its identification number to the $next_{id}$, if equals, it means that it has to take movement decision; and the entire exploration process is repeated by itself[29]. The details of the proposed approach are written as flow-chart and it is shown in Figure 4.

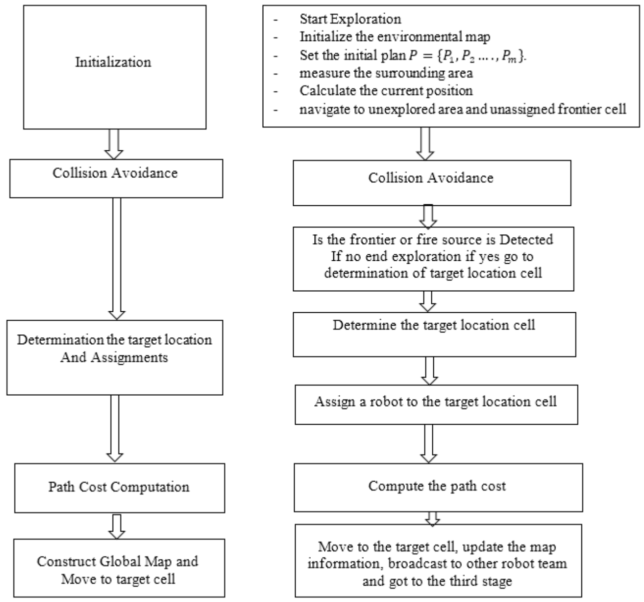

Figure 4 The flow-chart of the proposed approach

IV. Experimental Results

In The proposed approach is used for performing the environment exploration and fire source searching; using a team of heterogeneous robots of different sizes and with different sensing capabilities. A group of {4, 6, 8, 10} robots to study the performance of the suggested approach For all MRS teams two of them are equipped with pushing capabilities and the remaining are equipped with thermopile temperature sensor. To conduct these experiments, a simulation system is developed using a personal laptop; it allows observing the effects of different parameters on task performance. An occupancy grid environment of squared 100 X 100 cells with various room sizes and structures is used for this simulation as shown in Figure 2. The proposed algorithm is evaluated based on five different criteria that affect the performance of the task:

- The average number of movements or steps taken by an individual robot to accomplish the task using different team sizes of mobile robots.
- The length of the path taken by each individual robot to accomplish the task. It is identified by evaluating the planned trajectory of each robot.
- Average power consumption by a robot team which is one of the most important problems in robotics field since stopping and turning robots may consume a large amount of energy.
- Average number of turns taken by a robot team when it faces the crossover or obstacle, it must be as less as possible since it prevents robots from colliding with themselves or with obstacles which always preserve the energy consumption in all cases.

- The communication hops or number of hops taken to accomplish the task, various communication models are deployed and tested in the simulator; since the link bridges or paths that are established to connect a pair of robots may have multiple jumps or hops, each of them may raise some delay in the communication network between the robot team.

Figure 5 shows the number of steps or the number of movements that are required to complete the task when using different sizes of robot teams. The results show that the number of steps required when using a team of (4, 6, 8, 10) robots is (1250, 905, 753, 662) steps respectively. Therefore, the obtained results show that as MRS size increases, the required steps to accomplish the task will be decreased. Since the global map is decomposed into local sub-maps, each robot will effectively generate its local map within limited area; and this leads to the decomposition procedure taking less step number and less time.

Figure 6 presents the relation between the number of robots constitute the team size with the planned path taken to perform the process of environment exploration and fire searching; using different team sizes. The results show that the path length when using a team of (4, 6, 8, 10) robots is (1340, 980, 813, 703) respectively. It shows that the length of the path decreases while the number of robots increases; since it minimizes the number of movements to reach the nearest unexplored cell of the environmental map. Therefore, it improves the performance of the task for the selected environment.

To study the average power consumption, the team of MRS is studied for different numbers of mobile robots, the acceleration and deceleration caused by stopping and turning the robot may consume a large amount of energy because the path may have short distance but consumes more energy since the robot states (position and orientation) have different directions. Therefore, it is always preferable to have an efficient energy path with a moderate loss of distance[30][31]. The choice of the next cell depends not only on the distance but also on the direction of the robot movement. The robot's state is represented by its location and direction; it allows moving from one cell to any of its eight neighbors considering the energy for stops and turns; if the two states have different directions. But if two states have the same direction, the robot does not stop or turn; and in this case we assume energy consumption as zero.

Table 2 shows the energy consumption rate for different turns and stop. In Figure 7 there are three turns, moving from (A to B) requires 180° turn where A is the starting robot/node, from (B to C) requires 90° turn and finally from (C to D) where D is the target node, requires 45° turn. Considering the energy for stops and turns if two states have different direction, therefore, energy consumption of edge for example 'A to B' is 3.9 (0.9 +3), 0.9 for stop, and 3 for 180° turn. In a similar way, we calculate the energy consumption of the other edges. The solid arrow represents the current movement direction of the robot.

The function of recording the path is mathematically given by the following formula:

$$\nabla = C + E + C^* \tag{5}$$

Where, E is the consumption of energy from start to current cell, C is the obtained cost from start to current cell and C^* is the estimated cost from the current to target cell. Therefore, if the energy is much lower and the planned path is quite large, this path will always be the best[31].

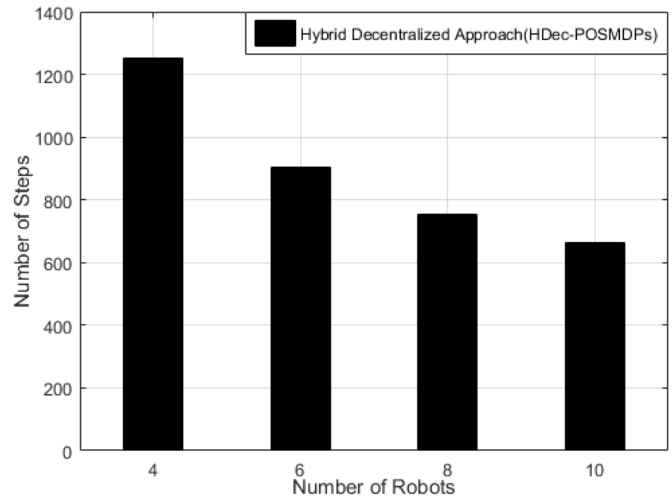

Figure 5 Number of robots vs number of steps

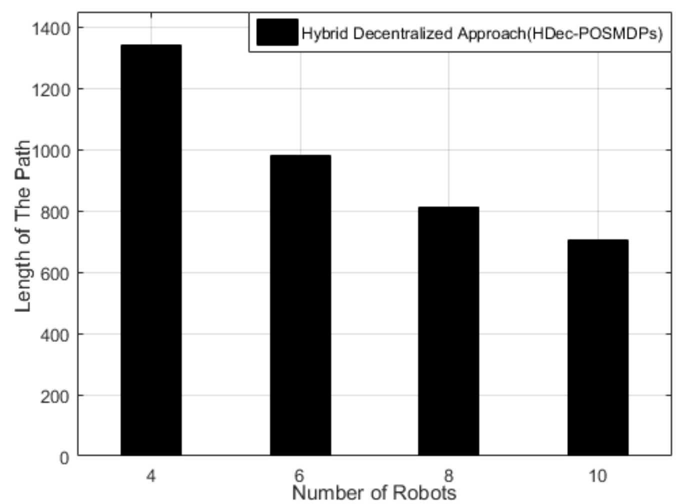

Figure 6 Number of robots team vs The path length

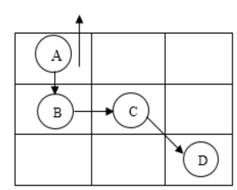

Figure 7 The performance parameter calculation example, the small circle represents robot's current location

Table 2 The consumption of energy rate for different turns and stop

Turns/Stop	Consumption of Energy
Stop	0.9
45°	1.2
90°	1.8
135°	2.4
180°	3

Figure 8 shows the energy consumed by different team size of robots when using the proposed approach for our problem, it shows that the energy consumed when using a team of (4, 6, 8, 10) robots is (298, 246, 196, 142) respectively; since it smoothes a path generated and it minimizes the energy consumption in an unnecessary turning of the robots.

Figure 9 plots a relation between the numbers of turns taken by each robots team to accomplish the task versus the number of robots constitute MRS team. The obtained results show that the number of count needed to accomplish the task when using the proposed approach decrease while the number of robot team increases. It shows that the number of counts when using a team of (4, 6, 8, 10) robots is (248, 199, 157, 104) respectively; since the decomposition of the map number of cycles is always lower since it prevents robots from colliding with themselves or with obstacles which always preserves the energy consumption in all cases.

Each robot must communicate with other robots to collect their local maps. The link bridges or paths that are established to connect a pair of robots may have multiple jumps or hops, each of them may raise some delay in the communication network between the robot team. Therefore, it will take longer time to transmit the message as the number of hops increases. The essential idea behind this problem is to reduce the total number of jumps to make an appropriate map of the environment at each network node.

A communication mechanism is created so that robots can share their local information with each other at each movement step. This mechanism is a communication protocol and it enables all robots of configuring a globally consistent map and planning their next movements in order to maintain network connectivity.

Robots are moving together to form a mobile network and share relevant information with the team. There are two types of communications (i) Centralized Communication System: robots are always required to maintain contact with a fixed base station. The central system gets solutions close to the optimal level; but has one point of failure, and (ii) Distributed Communications System: there is no base station and the communication is maintained through a wireless network. This paper concentrates on the distributed communications system to overcome the failure occurred through the central system.

Figure 10 shows the total number of jumps or hops needed by the proposed approach when using a team of 4 robots remains within a range of (77 to 113), for 6 robot team it remains within a range of (75 to 102), for 8 robot team it remains within a range of (61 to 98), and for 10 robot team it remains within a range of (53 to 86) which means a small number of hops to complete the task is required by the proposed approach; so it improves the performance of the task.

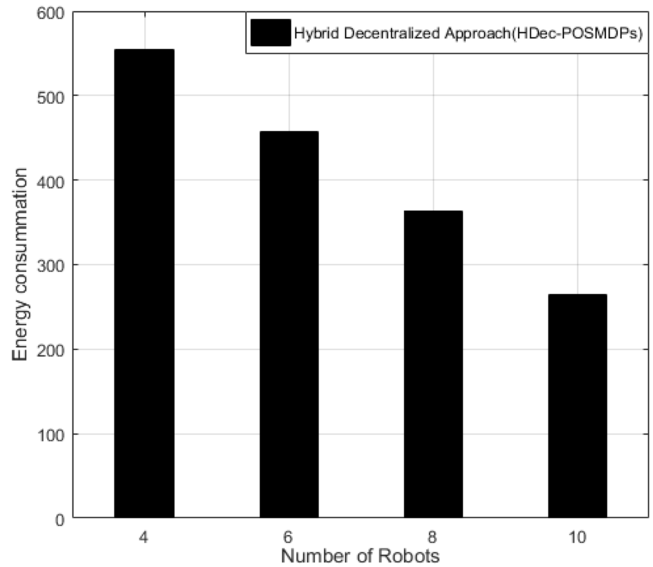

Figure 8 Number of robots vs The energy consummation

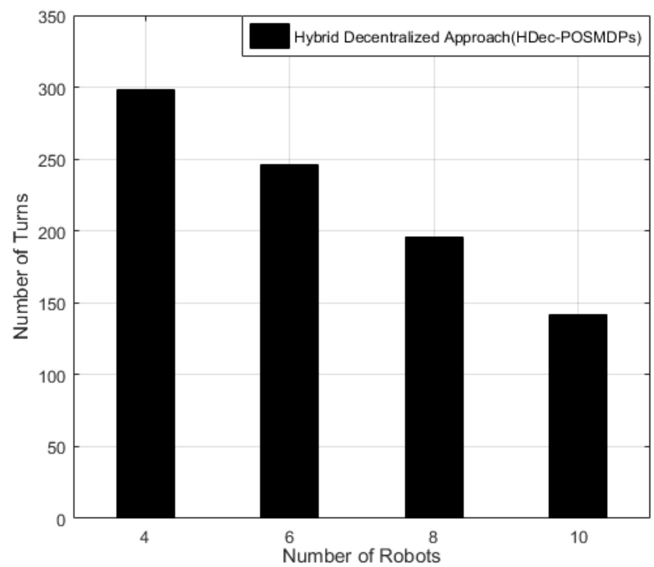

Figure 9 Number of robots vs Number of turns

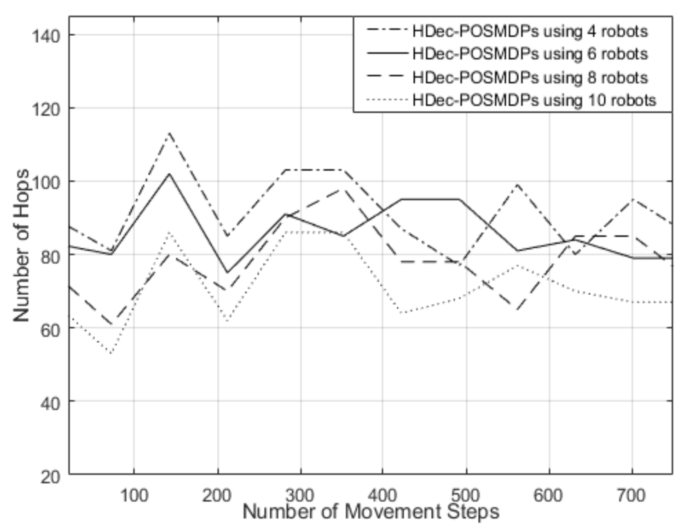

Figure 10 Number of steps vs Number of hops

V. Conclusions and Future Works

In this paper, a hierarchical top-down control architecture in multi-task decomposition methodology for coordination of MRS based on HDec-POSMDPs is proposed. Therefore; a team of heterogeneous robots with different sensory capabilities is considered, they cooperate to simultaneously explore different areas of the environment as well as detecting the source of the fire and reporting about it. A top-down design framework to a multi-robot scenario and iterative design approach is used to guarantee the global performance collectively. The paper aims at minimizing the overall mission time, making it possible to localize fire sources in an efficient manner, minimizing the energy consumed by each robot, minimizing the number of turns and reducing the number of hops in the networked robots. The global task is decomposed into multiple local sub-tasks which are assigned to one or more robots to be executed. The task is applicable to divide and conquer design to guarantee the global behavior of the mission. It is given as regular languages while the Multi-robot systems are modeled as a discrete-event system and each robot is represented by a deterministic finite state automaton model. The most effort in this paper is dedicated to the decomposition methodology, and ignores the complexity of the environment and the MRS size. This point can be manipulated in a future work. The areas explored by each robot and the quality also are two important problems that were not discussed in the paper. They will be addressed in the future.

References

[1] M. Al-khawaldah, T.M. Younes, I. Al-Adwan, M. Nisirat and M. Alshamasin, "Automated Multi-Robot Search for a Stationary Target," *International Journal of Control Science and Engineering*, vol. 4, no. 1, pp. 9-15, 2014.

[2] M.G. Earl and R. D'Andrea, "A decomposition approach to multi-vehicle cooperative control," *Robotics and Autonomous Systems*, vol. 55, no. 4, pp. 276-291, 2007.

[3] J.G.M. Fu, T. Bandyopad and M H.A. Jr, "Local Voronoi Decomposition for Multi-Agent Task Allocation," *IEEE International Conference on, Robotics and Automation* (ICRA), 2009.

[4] K. Hirayama, "A New Approach to Distributed Task Assignment using Lagrangian Decomposition and Distributed Constraint Satisfaction," *American Association for Artificial Intelligence* (www.aaai.org), 2006.

[5] A. Marjovi, J.G. Nunes, L. Marques and A.d. Almeida, "Multi-Robot Exploration and Fire Searching," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009.

[6] T. Gunn and J. Anderson, "Effective Task Allocation for Evolving Multi-Robot Teams in Dangerous Environments," *IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, 2013.

[7] M. Andriesand and F. Charpillet, "Multi-robot exploration of unknown environments with identification of exploration completion and post-exploration rendez-vous using ant algorithms," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[8] F.F. Carvalho, R.C. Cavalcante, M.A.M. Vieira, L. Chaimowicz and M.F.M. Campos, "A multi-robot exploration approach based on distributed graph coloring," *Latin American Robotics Symposium and Competition (LARS/LARC)*, 2013.

[9] A. Tsalatsanis, A. Yalcin and K.P. Valavanis, "Optimized Task Allocation in Cooperative Robot Teams," *17th Mediterranean Conference on Control and Automation*, 2009.

[10] G. Pini, A. Brutschy, C. Pinciroli, M. Dorigo and M. Birattari, "Autonomous task partitioning in robot foraging: an approach based on cost estimation," *Adaptive Behavior*, vol. 21, no. 2, pp. 118–136, 2013.

[11] H. Choi, A.K. Whitten and J.P. How, "Decentralized Task Allocation for Heterogeneous Teams with Cooperation Constraints," *American Control Conference (ACC)*, 2010.

[12] G.A. Korsah, A. Stentz and M.B. Dias, "A comprehensive taxonomy for multi-robot task allocation," *The International Journal of Robotics Research*, vol. 32, no. 12, pp. 1495–1512, 2013.

[13] B.P. Gerkey and M.J. Mataric, "A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems," *The International Journal of Robotics Research*, vol. 23, no. 9, pp. 939-954, 2004.

[14] A. Partovi and H. Lin, "Assume-guarantee Cooperative Satisfaction of Multi-agent Systems," *American Control Conference (ACC)*, 2014.

[15] J. Dai and H. Lin, "Automatic synthesis of cooperative multi-agent systems," *IEEE 53rd Annual Conference on Decision and Control (CDC)*, 2014.

[16] J. Dai, A. Benini, H. Lin, P. J. Antsaklis, M. J. Rutherford and K. P. Valavanis, "Learning-based Formal Synthesis of Cooperative Multi-agent Systems with an Application to Robotic Coordination," *24th Mediterranean Conference on Control and Automation (MED)*, 2016.

[17] M. Guo and D.V. Dimarogonas, "Bottom-up Motion and Task Coordination for Loosely-coupled Multi-agent Systems with Dependent Local Tasks," *IEEE International Conference on Automation Science and Engineering (CASE)*, 2015.

[18] J. Elston and E.W. Frew, "Hierarchical Distributed Control for Search and Tracking by Heterogeneous Aerial Robot Networks," *IEEE International Conference on Robotics and Automation*, 2008.

[19] J. Tumova and D.V. Dimarogonas, "A Receding Horizon Approach to Multi-Agent Planning from Local LTL Specifications," *American Control Conference (ACC)*, 2014.

[20] P. Schillinger, M. Bürger and D.V. Dimarogonas, "Decomposition of Finite LTL Specifications for Efficient Multi-Agent Planning," *13th International Symposium on Distributed Autonomous Robotic Systems, Cite this publication*, 2016.

[21] M. Karimadini and H. Lin, "Guaranteed global performance through local coordinations," Elsevier, Automatica, vol. 47, pp 890-898, 2011.

[22] X. Dai, L. Jiang and Y. Zhao, "Cooperative exploration based on supervisory control of multi-robot systems," *Springer, Applied Intelligence*, vol. 45, no. 1, pp. 18–29, 2016.

[23] S. Omidshafiei, A.A. Mohammadi, C. Amato, S. Liu, J.P. How and J. Vian, "Decentralized control of multi-robot partially observable Markov decision processes using belief space macro-actions," *The International Journal of Robotics Research*, vol. 36, no. 2, pp. 231–258. 2017.

[24] Y. Kantaros and M.M. Zavlanos, "Distributed Intermittent Connectivity Control of Mobile Robot Networks," *IEEE Transactions on Automatic Control*, vol. 62, no. 7, pp. 3109- 3121, 2016.

[25] M. Liu, K. Sivakumar, S. Omidshafiei, C. Amato and J. P. How, "Learning for Multi-robot Cooperation in Partially Observable Stochastic Environments with Macro-actions," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS*), 2017.

[26] V. Spirin and S. Cameron, "Rendezvous Through Obstacles in Multi-Agent Exploration," *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2014.

[27] J. Banfi, A.Q. Li, I. Rekleitis, F. Amigoni and N. Basilico, "Strategies for coordinated multirobot exploration with recurrent connectivity constraints," *Springer, Autonomous Robots*, vol. 42, no. 4, pp. 875–894,2017.

[28] J. Faigl, M.K and L. Preucil, "Goal Assignment using Distance Cost in Multi-Robot Exploration," *IEEE International Conference on Intelligent Robots and Systems*, 2012.

[29] A. Pal, R. Tiwari and A. Shukla, "Coordinated Multi-Robot Exploration under Connectivity Constraints," *Journal of Information Science and Engineering*, vol. 29, no. 4, pp. 711-727, 2013.

[30] J.d. Hoog, S. Cameron and A. Visser, "Role-Based Autonomous Multi-Robot Exploration," *Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns*, pp. 482-487, 2009.

[31] A. Pal, R. Tiwari and A. Shukla, "Multi-Robot Exploration in Wireless Environments," *Cognitive Computation Springer Science+Business Media*, vol. 4, pp. 526–542, 2012.