Goal-aware Team Affiliation in Collectives of Autonomous Robots

Lukas Esterle
Aston Lab for Intelligent Collectives Engineering (ALICE)
Aston University, Birmingham, UK
l.esterle@aston.ac.uk

***Abstract*—Collaboration in teams is essential in robot collectives. In order to achieve goals, individual robots would otherwise not be able to accomplish. In a such a distributed and highly dynamic system, a global coordination might not be possible. In this paper, we analyse static team affiliations, defined at deployment time, and compare its efficiency against dynamic team affiliations generated during runtime using random selection. Since operators might not be able to determine all dynamic aspects of the given environment at the time of deployment, we further propose a novel, goal-aware approach to affiliate each robot with a team. This approach brings together insights from biology, sociology, and psychology. In this novel approach, robots only operate on aggregated information from the network which is potentially changing during runtime. Finally, we also introduce an approach to select a team affiliation during runtime using machine learning techniques. In 60,000 experiments we analyse the efficiency and further discuss the different benefits and drawbacks of the proposed approaches.**

I. Introduction

In robot collectives, a set of autonomous robots interacts and potentially collaborates in order to achieve their individual goals. In many cases, these goals correlate and often collaboration in teams will yield in better and more efficient outcomes. Such team formations, whether temporal or permanent, become even more pressing when given tasks can not be accomplished by individual robots alone. Team formation has received a lot of attention in various research areas such as biology [1], [2], [3], sociology [4], [5] or robotics [6], [7], [8].

In this work, we are specifically interested in the self-organising ability of robots to affiliate themselves with a team at runtime, based on the current performance of the team towards its goals. Moreover, we investigate whether simple reactive behaviour, based on instantaneous information, or (machine) learning techniques, employing information learnt over a longer time horizon, is more efficient. We investigate this in a multi-robot setting with robots having only limited and directed areas for sensing/acting. Specifically we investigate the multi-object k-coverage problem [9]. The multi-object k-coverage problem has two main goals. On one hand, the network has to detect as many tasks in the environment as possible and on the other hand the collective has to provision a set of randomly arising tasks with k robots, i.e. k robots having the task within their sensing/actuating region [10].

This problem brings about an interesting trade-off where the network, when homogeneously working towards provisioning each task with k robots, they will inherently neglect large areas of the environment in which new tasks may arise and hence will not be able to detect those. In turn, if all robots try to cover the entire area, provisioning tasks with k or more robots can only happen if the initial layout does bring about the respectively required overlap in areas they can provision.

To overcome this problem, we define two distinct type of teams where each robot in a given team would follow specific goals: *observers* and *followers*. *Observer* try to maximise the number of detected tasks in the network by covering the initially given area. *Followers*, on the other hand, try to continuously provision tasks in order to ensure the task is k-covered.

Trying to balance this trade-off with robots affiliating themselves with teams, allows us to define the following research questions:

1) Can a permutation of multiple teams outperform single, homogeneous teams in terms of achieved k-coverage and the number of detected tasks?
2) Does a random selection of a team affiliation by the individual robots improve the performance of the entire network with respect to k-coverage and the number of detected tasks?
3) Can we enable each robot to make more profound decisions about team affiliation to improve network-wide performance in terms of k-coverage and the number of detected tasks?

The contributions of this paper are as follows. First, we analyse the benefit of having multiple teams in comparison to single, homogeneous teams within a robot collective. Second, we propose a goal-aware team affiliation approach and we show that such a dynamic affiliation with a team during runtime, based on network-wide aggregated information about the state of achieving the goal, can outperform static teams assigned at deployment. Furthermore, we compare this approach based on snapshot information against machine learning approach, building up knowledge over longer periods of time. Finally, we show that, while using the same performance metric, our proposed approach as well as learning-based approaches perform consistently even when changing the underlying goal of the collective.

The remainder of this paper is structured as follows. In the next section we give a formal problem description. In Section III we discuss related work in the field of dynamic team affiliation and task assignment as well as field of multi-object k coverage. Section IV analyses the benefits of heterogeneous teams over the performance of homogeneous teams and Section V discusses different dynamic team affiliation approaches, including a novel, distributed, goal-aware team affiliation approach. Section VI measures the efficiency of the proposed approaches using an additive (ϵ, δ)-approximation scheme. Section VII concludes the paper and gives an outlook on potential future work.

II. Problem Statement

The multi-task k-assignment problem is defined in [10] and can be stated as follows.

Consider a set of tasks $O_t = \{o_1, o_2, ..., o_m\}$ at time t that need to be provisioned and a set of mobile robots $R = \{r_1, r2, ..., r_n\}$. Importantly, the set of robots remains constant while the set of tasks may change over time as new tasks may arise and others disappear/resolve. Tasks are unknown to the robots until a task is within the sensing area of a robot. They arise and remain with a probability of σ and a duration of γ, respectively. Neither tasks nor robots are static and can move around in a 2D plain with a velocity $\vec{v}_i(t)$ and an orientation $\omega_i(t)$, an angle defining the orientation of the robot relative to a fixed reference point. Their location at time t is denoted as $\vec{x}_i(t) = (x_i, y_i)$. Specifically, the discrete-time behaviour of a robot r_i can be defined as

$$\begin{aligned}\vec{x}_i(t+1) &= \vec{x}_i(t) + \vec{v}_i(t) \\ \omega_i(t+1) &= \omega_i(t) + u_i(t) \end{aligned} \tag{1}$$

This velocities $\vec{s}_i$ and u_i are controlled by an internal autonomous software agent.

Robots communicate via message passing and employ a unit disk model to simulate wireless communication with a fixed range of c_i for simplicity. Finally, each robot has a dedicated sensing/actuating area f_i. We consider a task o_a to be provisioned by a robot r_i if the task is within the sensing/actuating area f_i, where $f_i = (\omega_i, \beta_i, s_i)$. Here, ω_i is the orientation, β_i, defines the width on either side of ω_i, and s_i is the range of the sensing area. Such directional sensing/acting areas can be found in robots with limited space for interaction or also in mobile smart camera systems. The range of a robots sensing/acting area is limited by the distance at which a task can be detected and provisioned. Therefore, a robot's state is defined as $r_i = \langle \vec{x}_i, \omega_i, f_i, c_i \rangle$. This defines a snapshot at a particular point in time and can be further indexed by t to represent the robot's state over time.

The mobile robots are now tasked with two goals. First, they should maximise the number of tasks being provisioned at any given time t. We consider a task o_a to be provisioned at time t by a robot r_i, if the task is geometrically within f_i:

$$prv(o_a, f_i, t) = \begin{cases} 1, & \text{if } d_{i,a} \leq s_i \;\&\; |\alpha_{i,a}| \leq |\beta_i| \\ 0, & \text{otherwise,} \end{cases}$$

where $d_(i,a)$ represents the distance to the task and $\alpha_{i,a}$ represents the angular distance between the orientation fo the robot and the task.

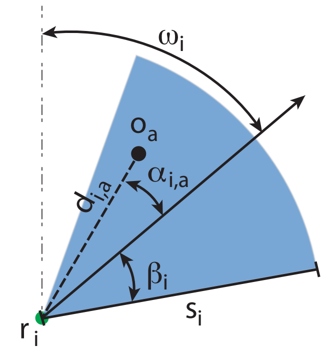


Fig. 1. Illustration of a task in a robot's sensing/actuating area. The sensing/actuating area is illustrated in blue with a range s_i, an orientation ω_i, and angle β_i on both sides of ω_i. The task is at angle $\alpha_{i,a}$ to the robot's orientation and distance $d_{i,a}$.

However, each task requires k robots to be accomplished. Therefore, the second goal requires them to maximise the number of task provisioned with k robots. We consider a task o_a to be provisioned by k robots at any given time t if

$$kprv(o_a, k, t) = \begin{cases} 1, & \textit{if } \sum_{i=1}^{n} prv(o_a, f_i, t) = k \\ 0, & \textit{otherwise.} \end{cases}$$

We can therefore define a normalised metric of how well online multi-task k-assignment is achieved as:

$$OMC_k = \sum_{t=1}^{T} \frac{\sum_{a=1}^{m} kprv(o_a, k, t)}{|O_t|} \tag{2}$$

for a given value of exactly k. Additionally, we define OMC_{k+} as the result where we consider all results of k but also incorporate all results where more than k robots provisioned the task.

Furthermore, we are interested in the number of detected tasks by the network:

$$DET = \sum_{t=1}^{T} \sum_{a=1}^{m} det(o_a, t) \tag{3}$$

where

$$det(o_a, t) = \begin{cases} 1, & \textit{if } \sum_{i=1}^{n} prv(o_a, f_i, t) \geq 1 \\ 0, & \textit{otherwise.} \end{cases}$$

III. Related Work

Team formation and respective team maintenance has been widely studied in nature and sociology [3], [11], [12]. In nature, social animals often form teams in order to overcome common goals, these goals can either be on the entire network-level (e.g. in foraging bees or ants) or on the team-level (e.g. in groups of spotted hyena). Besher and Fewell [3] thoroughly discuss models for division of labour which inherently lead to team formation within social insects. Conradt et al. discuss group decision-making in animals as well humans [1], [12] and how such decisions can lead to the formation of groups.

Gross et al. [6] discuss to what degree individuals, performing a collaborative task, require individual recognition or inter-individual differences in order to achieve complex division of labour in self-organised groups. Their findings show that teamwork requires neither individual recognition nor inter-individual differences among the participating entities. Pascal et al. [13] investigate different group sizes to achieve efficient team allocations to given tasks in foraging scenarios. Their presented models show that the number of allocated individuals per team can have an impact on the performance of the entire system in achieving their goals. Theraulaz et al. [14] investigate the benefits of reinforcing the threshold for selecting a team affiliation in foraging scenarios. They show how learning process for this threshold will lead to robust task allocations and specialisation of the individuals.

Processes of team building have also been heavily studied in humans in the area of Psychology, Sociology, and Management (e.g. [15]). Marks et al. propose a taxonomy of team processes and argue that they can generally divided into an action phase and a transition phase towards new actions. However, the different processes can be transient across phases. In contrast, we incorporate this transition phase entirely into the respective action phase. Standifer et al. [4] investigates the impact of temporal shared mental models on the performance of a collective to achieve a common task. They show that such a shared mental model can benefit not only the individual team but the collective of all teams if they are working towards a common goal.

Soon technical sciences picked up team formation especially in the area of multi-robot systems collaborating towards a new goal. Jennings et al. [8] propose to use teams of mobile robots for collaborative search and rescue operations. Individual robots would search the area and upon encountering an object, they will request others to help 'rescue' the object. Krieger et al. [16] employ techniques observed in ant foraging processes to allocate tasks and recruit new robots into the team. They found that affiliating entities with teams based on information exchange are more efficient in achieving their task than without information exchange. They also show the impact of the team size on the performance of the respective team. Bererton et al. [7] enables teams of robots to collaborate in repair and docking tasks. They envision such systems of collaborative robotic teams to be deployed on deep space exploration missions. Grabowski et al. [17] uses autonomous robots for exploration tasks. Here the robots can join different teams in order to explore different areas of the environment. Furthermore, they collaborate in order to improve their localisation capabilities.

Another important application in multi-robot systems is called *Cooperative Multi-robot Observation of Multiple Moving Targets (CMOMMT)*. Here a set of robots tries to find and cover moving targets that are initially unknown. The problem was first introduced by Parker and Emmons [18]. Werger and Matarić [19] propose W-CMOMMT (weighted CMOMMT) giving a weight to each target. A robot then broadcasts local eligibility in order to coordinate tasks among all robots. Since targets are initially unknown, Jung and Sukhatme [20] learn densities of sensors and targets. They use this information to direct idle robots to insufficiently covered areas. To ensure continuous 1-coverage of various objects, Kollin and Carpin [21] perform a target loss prediction allowing the individual devices to call for help in a timely manner.

In the multi-object k-coverage problem, studied in this paper, CMOMMT is combined with the k-coverage problem of sensor networks introduced by Huang and Tseng [22]. The k-coverage problem assumes a set of sensors covering the entire environment. The goal is to select a subset in order to cover specific stationary locations with at least k sensors. This allows to conserve resources of the network by turning off sensors not in the required subset. Hefeeda and Bagheri [23] propose a distributed approach to approximate optimal k-coverage in a network. Elhoseny et al. [24] propose using mobile nodes to cover known and stationary targets with k sensors. In order to optimise the coverage, they use an evolutionary approach. Liu et al. [25] analyse the benefits of moving sensors to detect and cover specific, but unknown, stationary points in the environment. Fusco and Gupta [26] propose a simple greedy algorithm to optimally place and orient directed sensor for k-coverage of static objects in the environment. In camera networks, Micheloni et al. [27] identify activity density maps, determining areas highly frequented by target objects and use an expected-maximization process to define optimal orientations of PTZ cameras. CMOMMT and coverage optimisation in camera networks has been researched quite intensively [28], [29], [30]. However, the problem of k-coverage with unknown number of mobile targets only received little interest yet.

IV. Static Team Affiliation

In order to achieve high number of provisioned tasks OMC_k and OMC_{k+} as well as a high number of detected tasks DET, Esterle and Lewis [9] propose different communication and response models. However, Esterle notes that static devices, not moving in the environment, detect the highest number of tasks in comparison to any of the other approaches employing mobile devices [10], [31]. Therefore we want to analyse the effect of having mobile and static (immobile) robots on the performance of the collective. We consider two fundamentally different type as separate, but collaborative, teams to which each robot can join:

- *Observer*: are robots that *observe* the environment from their initial position and given their initial pose. They provision only those tasks passing through their sensing area.
- *Follower*: are robots that can move around in the environment. They actively *follow* a designated task and abandon their initial position.

A robot r_i can detect tasks autonomously as soon as the task is within the f_i. The robot continuously broadcasts the position of the task to other robots in the network as long as it is within f_i. Each robot will set the closest task as its designated task. An autonomous internal software agent will try to keep the task within f_i by adjusting $\vec{v}_i$ and u_i accordingly.

Importantly, each type of team has its benefits and drawbacks with respect to the multi-task k assignment problem for the entire network. On one hand, *observer* teams remain in their initial position and cannot follow tasks through the environment. This severely limits the duration of provisioning the respective task. Furthermore, this means, provisioning tasks with multiple robots can only occur when these robots have overlapping sensing/acting areas to begin with. However, *observers* can cover larger areas and detect more tasks in the environment overall. On the other hand, *follower* teams can provision tasks over longer periods of time and can achieve k coverage more easily as they can re-locate and are not restricted to their initial location for acting/sensing. However, *follower* moving around might lead to clustering of robots in certain locations making the remaining area of the scenario not covered. This means, tasks appearing in these uncovered areas are missed completely and are not noticed at all.

First, we are interested in networks where a team is assigned statically to each robot and cannot be changed during runtime. This results in 2^n different static assignments or configurations in terms of team affiliations for a network with n robots. Second, we also analyse dynamically changing team affiliation where each robot (a) selects a team affiliation randomly, (b) uses our novel, goal-aware team affiliation scheme, and finally (c) uses well known machine learning approaches, i.e. multi-armed bandit problem solvers, to select and change its team affiliation at runtime.

A. Simulation Environment

We tested a homogeneous and heterogeneous assignment of mobile and static team affiliations across all robots on 10 randomly generated scenarios using CamSim [32]. All scenarios have an environment of the size 40×40 meter with 10 deployed robots. While robots are randomly placed and oriented, the size of their sensing area the same for all robots, i.e. an angle of $35°$ degrees ($\beta_i = 17.5$) and a range $s_i = 10$ meter. The goal of the network is to provision each task with $k = 3$ robots. Initial location and orientation of each robot has been sampled randomly from a uniform distribution. A total number of 5 tasks can be present at maximum at any discrete time step. Tasks and robots follow a random trajectory until reaching the border of the simulation environment and bounce back in a random trajectory ensuring a constant number of robots in the experimental environment. Tasks can not move faster than robots making sure *follower* robots can provision moving tasks continuously.

Our first results presented in Figure 3 gives an example outcome of scenario 6 and clearly shows the benefit of heterogeneous teams in the network, as expected. The experiment has been repeated 30 times due to stochasticity and mean values are shown. The red $\times$ show the performance of single homogeneous teams, i.e. all robots are either *follower* or *observer*. In our results we compare on one hand the total number of tasks discovered and the number of tasks provisioned. In all our initial experiments, pure *observer* teams perform better than homogeneous *follower* teams in terms of detecting tasks to be provisioned. At the same time, *observer* teams outperform *follower* teams when it comes to provisioning tasks with k robots. This clearly shows the trade-off between detecting a high number of tasks and provisioning them with k robots. Surprisingly, it is quite important whether we are interested to provision tasks with at *least* k robots or with *exactly* k robots. Figure 3c shows results where we were analysing the outcome of provisioning tasks with k or more robots, while Figure 3d shows the results where we analyse the results of provisioning each task with exactly k robots. While in both cases heterogeneous teams perform best, the ratio of *observer* and *Follower* has an important impact on the performance. If we are interested in provisioning tasks with at least k robots, a high ratio of *followers* is important (about $80 - 90\%$). In contrast, when aiming to provision each task with *exactly* k robots, a higher number of *observers* is better (again about $80 - 90\%$).

The number of detected tasks by each configuration in the scenario is normalised by the maximum number of tasks detected across all configuration. The number of tasks provisioned by k robots is normalised by the total number of available task, independent of whether they have been detected or not. Each of the 10 scenarios have been repeated 30 times and mean values are given.

While we can clearly see from our example results that a ratio of 80% *observers* leads to quite good outcomes, the spread in results with this ratio also indicates that the selected robot to affiliate with the *follower* team has an important impact.

V. Dynamic Team Affiliation

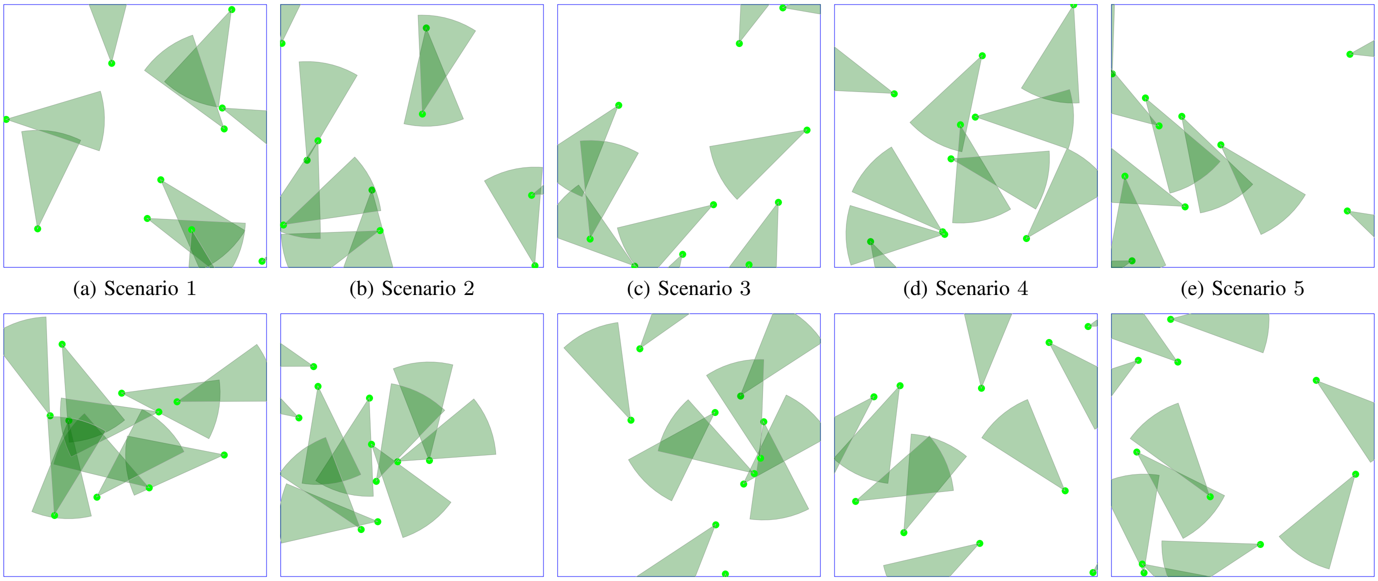
Instead of static assignment of team affiliations, we also investigate dynamic team affiliations within the collective. This means, each individual robot can decide whether it prefers to remain in its current team or switch to the other team at runtime. When switching to the observer team, a robot returns to its initial position. In this work, we investigated three different schemes to making such a decision: using a random decision process, a process based on collective goal-aware decisions, and a process based on machine learning techniques.

A. Random Team Affiliation

When employing *Random Team Affiliation (RTA)* each robot can select a new team at random. However, we do not allow each robot to select a team at any given time as this might lead to a strong fluctuation of team members in both teams. In other words, it might happen that a team member constantly switches team affiliations. To avoid this, we assume a robot commits itself for a given period of time Δ. During this period of time, a robot will not attempt to change its affiliation to another group. A team affiliation is randomly sampled from a uniform distribution.

B. Goal-aware Team Affiliation

As an alternative to RTA, we propose a novel approach enabling each robot to aggregate network-wide information



(a) Scenario 1 (b) Scenario 2 (c) Scenario 3 (d) Scenario 4 (e) Scenario 5

(f) Scenario 6 (g) Scenario 7 (h) Scenario 8 (i) Scenario 9 (j) Scenario 10

Fig. 2. Examples of scenarios: Green spots indicate the locations of the robots, associated green circle segments indicate their respective sensing areas.

about the current state of the collective and hence allowing it to make an informed decision about its team affiliation. The state of the network is not kept centrally but aggregates the state of the collective locally on each robot, using the broadcasted information of each robot in the network. While teams follow common goals (i.e. increase the number of detected tasks (DET) or increase the number of provisioned tasks OMC_k), the individual robots are not aware of which other robots belong to which team or how many members each team has.

Based on the disseminated information of each individual robot in the network, each robot can derive the following information. First, the number of tasks detected and currently within any f_i of the network

$$ndet(t) = \sum_{a=1}^{m} det(o_a, t), \tag{4}$$

and second, the number of robots currently provisioning at least one task

$$ndprv(t) = \sum_{i=1}^{n} dprv(f_i, t), \tag{5}$$

where

$$dprv(f_i, t) = \begin{cases} 1, & \textit{if } \sum_{a=1}^{m} prv(o_a, f_i, t) \geq 1 \\ 0, & \textit{otherwise}. \end{cases}$$

Employing this knowledge allows them to make a decision whether or not robots might be need to provision all available tasks. Since all robots receive the same information at potentially the same time, the robots use this information as a dynamic threshold for randomly sampled variable from a uniform distribution. The state is determined as follows:

$$state(t) = 1 - \left(\frac{\left(\frac{ndprv(t)}{k}\right)}{ndet(t)} \right)$$

where $state$ represents the required ratio of *followers* for the current number of tasks. This means, the state gives each individual robot a notion of required followers and observers in the team respectively. Therefore, depending on the value of state, each robot either affiliates itself with the team

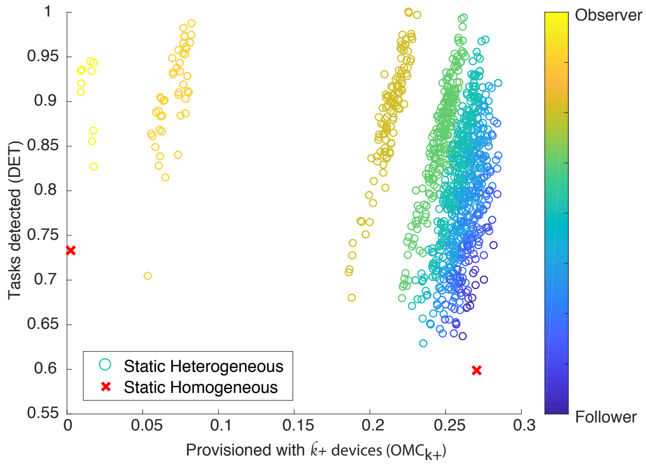
$$\begin{array}{ll} \textit{Observer}, & \textit{if } state(t) < 0 \textit{ and } |state(t)| \geq ran, \text{ or} \\ \textit{Follower}, & \textit{if } state(t) > 0 \textit{ and } |state(t)| \geq ran. \end{array}$$

where ran is a randomly sampled variable from an uniform distribution. A robot remains in its current team otherwise. We term this approach *Goal-aware Team Affiliation (*GTA*)*, as individual are aware of the goals within the network and have an idea of the progress towards achieving these goals on a network-level.

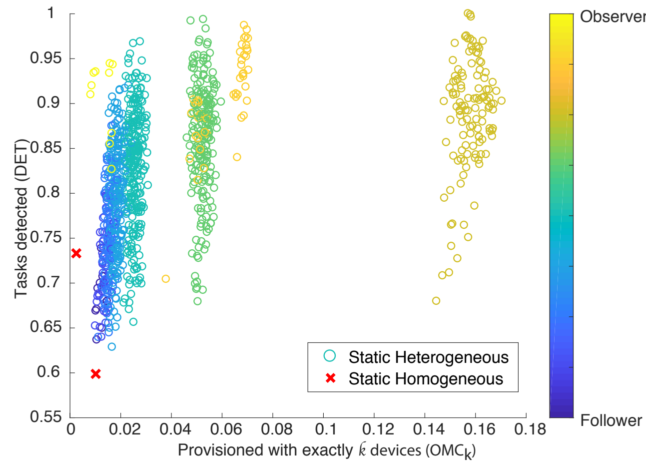
C. Team Affiliations using Machine Learning

In addition to GTA, we are also interested in using machine learning approaches to decide on a team affiliation at runtime by each individual robot (*Machine Learning Team Affiliation (MLTA)*). To do so, we employ simple multi-armed bandit-problem solvers from the literature, namely, ϵ-*Greedy* [33], *UCB1* [34], and *SoftMax* [33]. Multi-armed bandit problem solver idealise the explore vs. exploit dilemma, i.e. whether to stick with the current team or switch team affiliations. To provide equal opportunities to the bandit problem solving algorithms, we use the same information as for the GTA approach. However, instead of using $state$ as a dynamic threshold to decide whether to switch teams or not, we use it as part of the reward function of the current team for the multi-armed bandit problem solvers. For the teams we calculated the reward as follows:

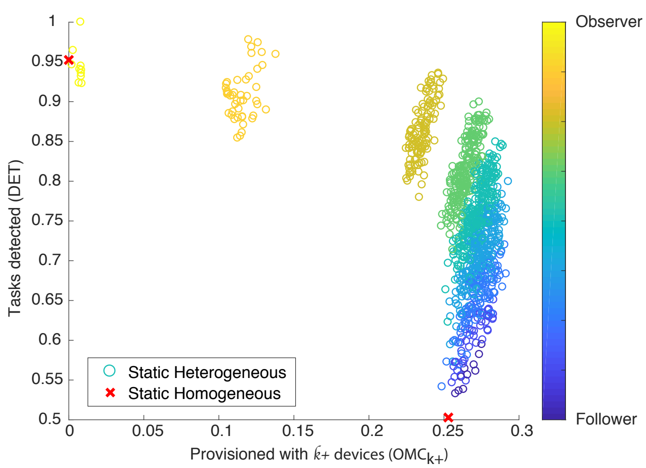
$$reward = (state(t)) \times \lambda.$$



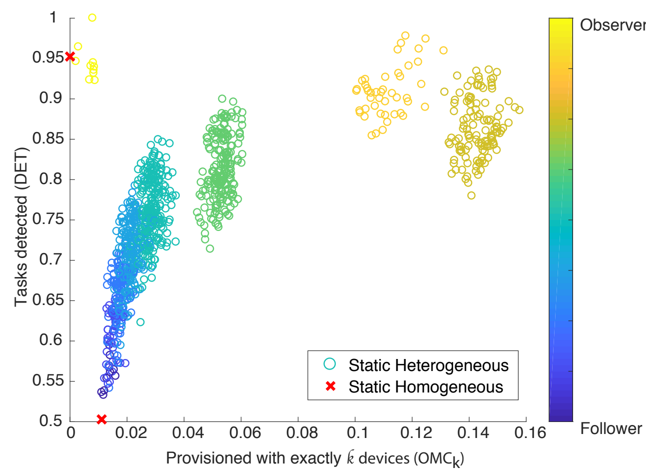
(a) Scenario 1: Results for provisioning with k or more robots



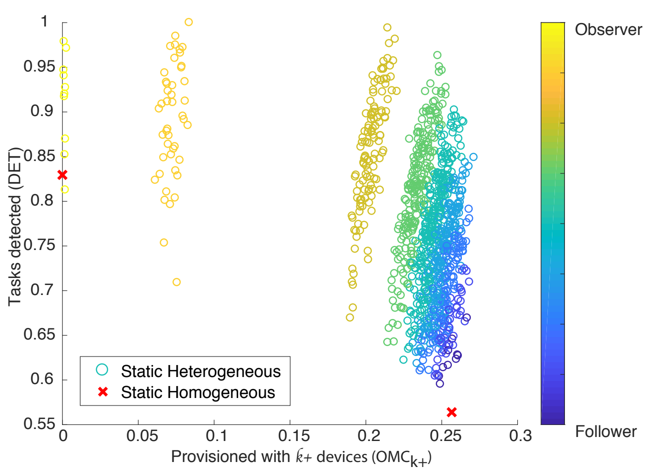
(b) Scenario 1: Results for provisioning with exactly k robots



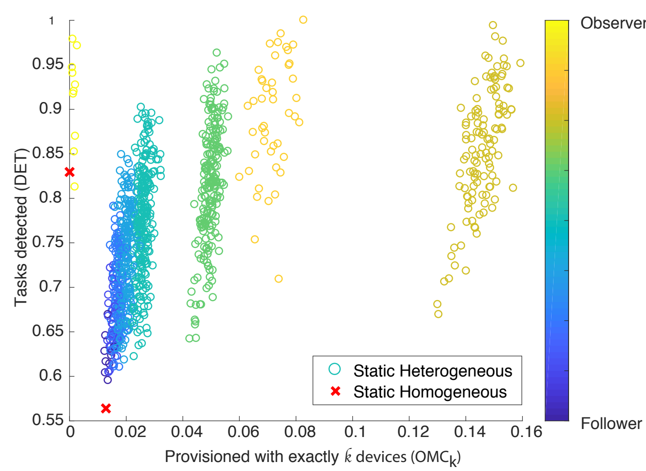
(c) Scenario 4: Results for provisioning with k or more robots



(d) Scenario 4: Results for provisioning with exactly k robots



(e) Scenario 10: Results for provisioning with k or more robots



(f) Scenario 10: Results for provisioning with exactly k robots

Fig. 3. Comparison of performance of homogeneous and heterogeneous teams. Figures on the left show the performance when trying to provision each task with k or more robots. Figures on the right hand side show the performance of heterogeneous and homogeneous teams when trying to provision each task with exactly k robots. The different colours for heterogeneous assignment indicate the ratio between the number of observer and follower robots.

where

$$\lambda = \begin{cases} 1, & \textit{if } d_i \textit{ is currently affiliated with follower} \\ -1, & \textit{otherwhise}. \end{cases}$$

At regular intervals, each robot will calculate the reward for its currently selected team locally and decide whether to affiliate with the other team or remain in its current team.

VI. Results

To test our dynamic team affiliation approaches, we first performed and compared RTA, GTA, and MLTA, using multi-armed bandit problem solver against (i.e. SoftMax [33], UCB1 [34] and ϵ-Greedy [33]), against *Static Team Affiliation* approaches in our 10 scenarios as depicted in Figure 2 using the simulation tool CamSim [32]. For *Dynamic Team Affiliations* (i.e. RTA, GTA, MLTA) each robots is initially affiliated with a random team sampled from a uniform distribution. SoftMax is used with a temperature of $\tau = 0.1$, determining the uniformity with which an action is selected. A low temperature means an action with an expected high reward is selected with a higher probability. ϵ-Greedy is used with $\epsilon = 0.1$ indicating the probability to explore a new team rather than exploiting the currently best one.

Figure 4 shows the mean performance of the different configurations and approaches averaged over 30 runs for each scenario with respect to DET and COV_k or COV_{k+}, respectively. For all experiments, $k = 3$, $\beta_i = 17.5$, $r_i = 10$, $\sigma = 0.05$ and γ is randomly drawn from $[5, 100]$ with a uniform distribution. For RTA, $\Delta = 20$ discrete time steps.

From the three examples given, we can see that in all cases UCB1 achieves the highest rate of detected task (DET)in the network throughout the entire scenario. In many cases, GTA is still able to outperform almost all heterogeneous team affiliations. Furthermore, we can note how GTA, RTA, and UCB1 outperform any static team affiliations in terms of provisioning tasks with at least k robots (OMC_{k+}). Importantly, we tested all potential affiliation configurations within a network and only a few of those are able to outperform GTA, RTA and UCB1 in terms of provisioning tasks with exactly k robots (OMC_k). We speculate that the initial location of each robot as well as the relation to other robots may indicate the optimal team for each robot. However, without *a priori* knowledge of the scenario, the optimal team for each robot, ensuring the network to outperform any dynamic team affiliation, is not determinable. Importantly, this *a priori* knowledge is not required for our proposed approach.

Tables I, II, and III present a pair-wise comparison of the different approaches, for individual robots to affiliate with teams, against each other. We tested the different approaches in a number of scenarios with arbitrary initial positions for the robots. The initial position of a robot also corresponds to the location and orientation an *observer* would take up.

Given the good outcomes of our dynamic team affiliation approaches, we would like to demonstrate that the number of experiments performed is sufficient for high confidence in our results. This requires us to determine the appropriate number N of random variables $Z_1, ...Z_N$ necessary for the Monte-Carlo approximation scheme we apply to assess efficiency of our approaches. For this purpose, we use the additive approximation algorithm as discussed in [35], [36]. If the sample mean $\mu_Z = (Z_1 + \ldots + Z_N)/N$ is expected to be large, then one can exploit the Bernstein's inequality and fix N to $\Upsilon \propto ln(1/\delta)/\varepsilon^2$. This results in an *additive* or *absolute-error* (ε, δ)*-approximation scheme*:

$$\mathbf{P}[\mu_Z - \varepsilon \leq \widetilde{\mu}_Z \leq \mu_Z + \varepsilon)] \geq 1 - \delta,$$

where $\widetilde{\mu}_Z$ approximates μ_Z with absolute error ε and probability $1 - \delta$.

In particular, we are interested in Z being a Bernoulli random variable indicating whether a specific approach outperformed another approach in terms of the number of detected tasks DET and the provision of tasks OMC_k and OMC_{k+}.

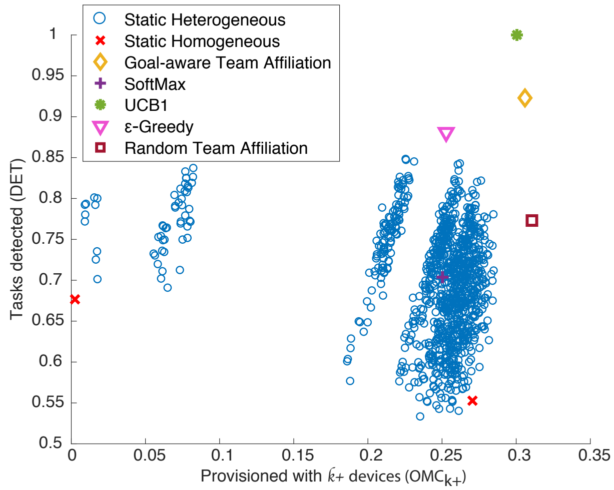
We can use the Chernoff-Hoeffding instantiation of the Bernstein's inequality, and further fix the proportionality constant to $\Upsilon = 4\, ln(2/\delta)/\varepsilon^2$, as in [37]. Hence, for our performed 60,000 experiments, we achieve the following success rates as given in the Tables I, II, III. The indicated results are a direct comparison of the different approaches and are given with an absolute error of $\varepsilon = 0.02$ and confidence ratio 0.99. Table I gives an overview of the comparison between the different approaches when trying to provision tasks with *exactly* k robots. While UCB1 and ϵ-Greedy perform better than pure *Follower* teams, our proposed GTA performs better than any of the other tested techniques. In this case, we ensure to not over-provision the tasks with robots. This is important when such over-provisioning does not yield in more efficient performance.

Interestingly, when the opposite is important, i.e. tasks do benefit from having more than k robots provision them, a simple *Random* assignment works best, as shown in Table II. Also, in not a single case of the performed 60,000 experiments, homogeneous *observer* teams outperformed any other approach with respect to provision tasks with $k+$ robots. However, in a direct comparison, GTA is usually quite close (in most cases within 5%) to the success rate of *Random*. Furthermore, GTA still outperforms *Random* in $\sim 41\%$ of all performed experiments.

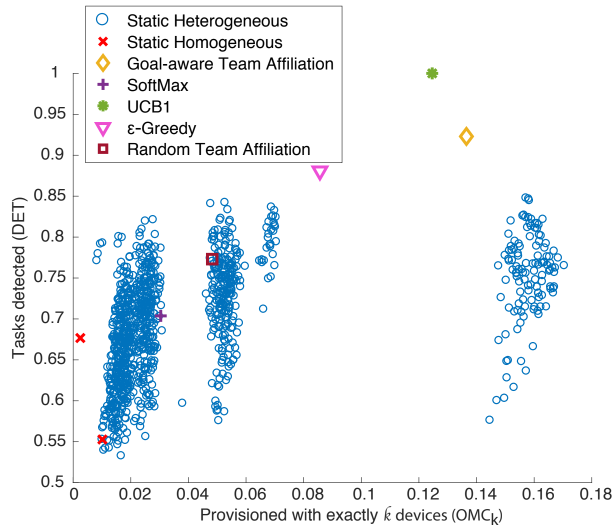
Finally, when comparing the performance with respect the number of detected tasks (cp. DET in equation 3) in Table III, we can see that throughout all experiments UCB1 can outperform the other approaches most often. Again, GTA is quite close in achieving similar success rates as UCB1. However, this result is rather unexpected as the fitness function for the bandit problem solver does not consider the number of detected tasks explicitly but rather implicitly limits the number of robots provisioning tasks and hence limits over-provisioning.

VII. Conclusion

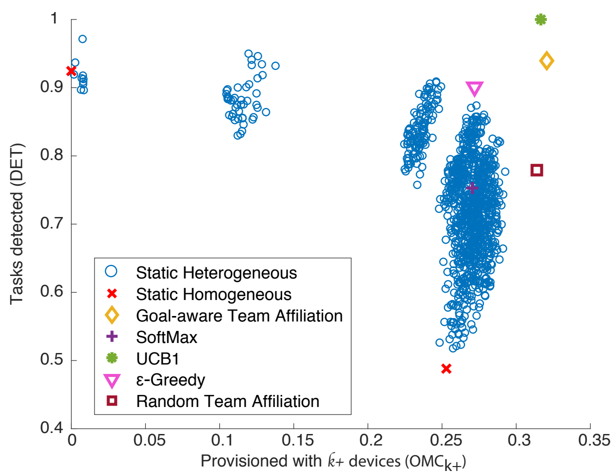
In this paper we investigated the effect of teams on the performance of multi-robot systems to achieve contrasting goals. The *follower* team follows tasks in order to provision them with at least k robots. This eventually leads to clusters of



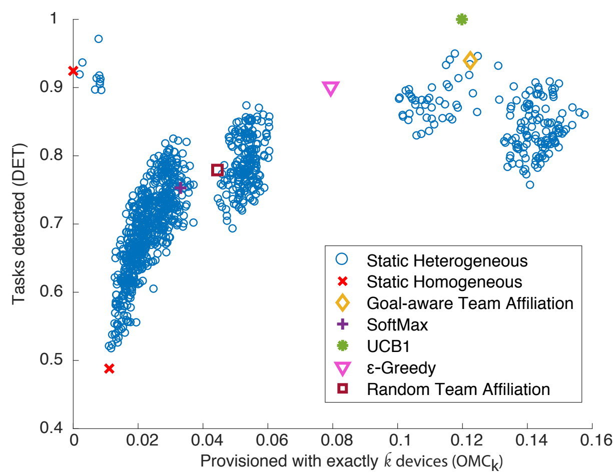
(a) Scenario 1: Results for provisioning with k or more robots



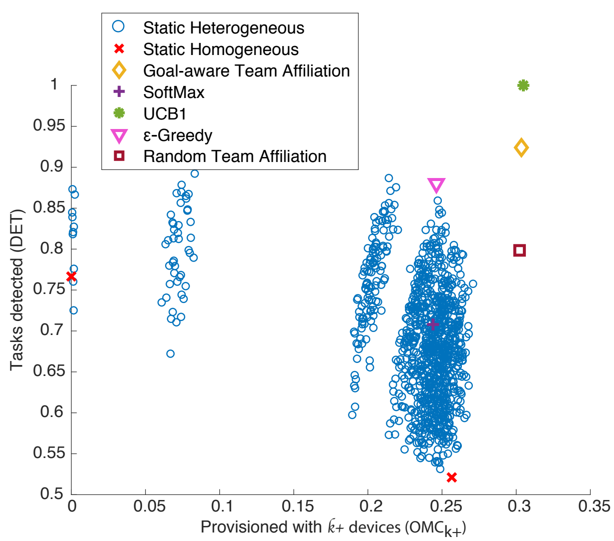
(b) Scenario 1: Results for provisioning with exactly k robots



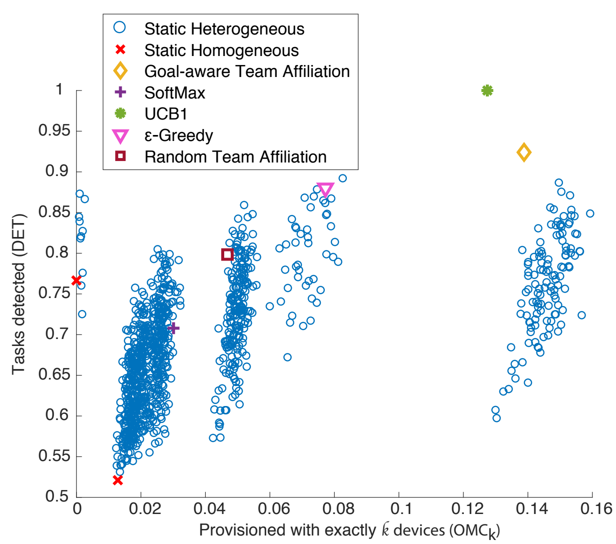
(c) Scenario 4: Results for provisioning with k or more robots



(d) Scenario 4: Results for provisioning with exactly k robots



(e) Scenario 10: Results for provisioning with k or more robots



(f) Scenario 10: Results for provisioning with exactly k robots

Fig. 4. Comparison of performance of homogeneous and heterogeneous teams against dynamic team affiliation during runtime. Figures on the left show the performance when trying to provision each task with k or more robots. Figures on the right hand side show the performance of heterogeneous and homogeneous teams when trying to provision each task with exactly k robots. The different colours for heterogeneous assignment indicate the ratio between the number of static and mobile robots.

	Follower	Observer	GTA	SoftMax	UCB1	ϵ-Greedy	Random
Follower	-	9.15	**100.00**	91.00	**100.00**	**100.00**	99.48
Observer	90.40	-	**100.00**	98.37	**100.00**	99.99	99.68
GTA	0.00	0.00	-	0.11	**34.23**	1.50	0.01
SoftMax	8.51	1.56	**99.89**	-	99.82	98.74	78.64
UCB1	0.00	0.00	**65.15**	0.18	-	2.70	0.02
ϵ-Greedy	0.00	0.01	**98.45**	1.23	97.21	-	4.51
Random	0.49	0.30	**99.99**	20.86	99.98	95.33	-

TABLE I

COMPARISON OF THE PERFORMANCE OF APPROACHES IN PERCENT WITH RESPECT TO THEIR ABILITY TO PROVISION TASKS WITH **EXACTLY** k ROBOTS. E.G. 98.58% OF THE EXPERIMENTAL RESULTS USING SOFTMAX ARE BETTER THAN THOSE RESULTS USING OBSERVER ASSIGNMENT. THE DIFFERENCE BETWEEN THE SUM OF TWO APPROACHES AND 100 REPRESENT THE PERCENTAGE OF RESULTS BEING EQUAL IN PERFORMANCE. A TOTAL OF 60,000 EXPERIMENTS HAVE BEEN PERFORMED. RESULTS HAVE A CONFIDENCE LEVEL OF 99% AND $\varepsilon = 0.02$ ABSOLUTE ERROR.

	Follower	Observer	GTA	SoftMax	UCB1	ϵ-Greedy	Random
Follower	-	0.00	84.59	36.78	82.04	37.72	**89.79**
Observer	**100.00**	-	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
GTA	15.22	0.00	-	5.44	42.41	3.87	**58.13**
SoftMax	62.84	0.00	94.44	-	92.60	50.55	**96.59**
UCB1	17.74	0.00	57.04	7.27	-	4.68	**63.39**
ϵ-Greedy	61.93	0.00	96.03	48.97	95.21	-	**97.34**
Random	10.04	0.00	**41.44**	3.34	36.19	2.59	-

TABLE II

COMPARISON OF THE PERFORMANCE OF APPROACHES IN PERCENT WITH RESPECT TO THEIR ABILITY TO COVER TASKS WITH $k \geq 3$ ROBOTS. E.G. 36.64% OF THE EXPERIMENTAL RESULTS USING SOFTMAX ARE BETTER THAN THOSE RESULTS USING FOLLOWER ASSIGNMENT. THE DIFFERENCE BETWEEN THE SUM OF TWO APPROACHES AND 100 REPRESENT THE PERCENTAGE OF RESULTS BEING EQUAL IN PERFORMANCE. A TOTAL OF 60,000 EXPERIMENTS HAVE BEEN PERFORMED. RESULTS HAVE A CONFIDENCE LEVEL OF 99% AND LESS THAN 2% ABSOLUTE ERROR.

	Follower	Observer	GTA	SoftMax	UCB1	ϵ-Greedy	Random
Follower	-	89.26	99.61	92.08	**99.66**	99.35	99.23
Observer	10.68	-	89.96	39.40	**95.78**	83.57	61.71
GTA	0.39	9.95	-	4.70	**74.82**	36.14	13.66
SoftMax	7.84	60.40	95.27	-	**97.83**	93.79	75.53
UCB1	0.34	4.16	**24.97**	2.14	-	17.26	6.05
ϵ-Greedy	0.65	16.31	63.59	6.14	**82.61**	-	22.99
Random	0.76	38.14	86.21	24.28	**93.89**	76.89	-

TABLE III

COMPARISON OF THE PERFORMANCE OF APPROACHES IN PERCENT WITH RESPECT TO THEIR ABILITY TO DETECT TASKS IN THE ENVIRONMENT. E.G. 88.89% OF THE EXPERIMENTAL RESULTS USING OBSERVER ARE BETTER THAN THOSE RESULTS USING FOLLOWER ASSIGNMENT. THE DIFFERENCE BETWEEN THE SUM OF TWO APPROACHES AND 100 REPRESENT THE PERCENTAGE OF RESULTS BEING EQUAL IN PERFORMANCE. A TOTAL OF 60,000 EXPERIMENTS HAVE BEEN PERFORMED. RESULTS HAVE A CONFIDENCE LEVEL OF 99% AND LESS THAN 2% ABSOLUTE ERROR

devices in specific areas of the environment and leaves other areas unattended. In contrast, the *observer* team take up a defined position in order to detect newly arising tasks in the environment. This leads to a trade-off between goals of which both of them should be maximised within the collective of both teams.

While assigning a static team to each of the robots in the network at deployment time already outperforms single, homogeneous teams, operators have to know exactly which device to select for which team. Furthermore, we showed in this work how static teams are affected by the specifics of the overall goal. Even slight variations in the goal formulation already result in drastically diverging outcomes in performance of the entire network.

To overcome this limitation, we enable devices to choose their own affiliation with a team locally and at runtime. By enabling robots to aggregate network-level goal information and use this to select a team at runtime, they can outperform randomly sampled affiliations. We further investigate and compare this with well known machine learning techniques, namely multi-armed bandit problem solvers. In 60,000 experiments we directly compared each individual approach against each other. While both, UCB1 and our novel approach, GTA, performed better than the remaining approaches, their strength depends on the actual goal they pursuit. While GTA performs best in provisioning tasks with exactly k devices, UCB1 constantly finds most of the tasks arising in the network.

In our future work we want to focus on two important aspects we have uncovered in this work. First, we have seen in a small set of experiments that heterogeneously selected team affiliations at deployment can outperform dynamic team affiliations in terms of provisioning tasks with exactly k devices. More thorough analysis of the properties that constitute and allow optimal affiliations at deployment time is required. This would not only allow further analysis of statically assigned teams at deployment against dynamic team assignments during runtime, but might also give further insight to improve dynamic team affiliations.

Finally, in this work we assume all robots are cooperative and support the respective team efforts. While non-collaborative robots should not pose a problem to the remainder of the network, we would expect that actively adversarial devices might indeed be problematic to the network-wide outcomes. In a next step, we will investigate the performance of the network in the presence of such adversarial robots, actively trying to perturb the network.

ACKNOWLEDGEMENT

This work was supported by the SOLOMON project (Self-Organisation and Learning Online in Mobile Observation Networks) funded by the European Union H2020 Programme under grant agreement no 705020.

REFERENCES

[1] L. Conradt and T. J. Roper, "Group decision-making in animals," *Nature*, vol. 421, pp. 155–158, 2003.
[2] I. Couzin, "Collective minds," *Nature*, vol. 445, p. 715, 2007.
[3] S. N. Beshers and J. H. Fewell, "Models of division of labor in social insects," *Annual Review of Entomology*, vol. 46, no. 1, pp. 413–440, 2001.
[4] R. Standifer and A. Bluedorn, "Alliance management teams and entrainment: Sharing temporal mental models," *Human Relations*, vol. 59, no. 7, pp. 903–927, 2006.
[5] M. A. Marks, J. E. Mathieu, and S. J. Zaccaro, "A temporally based framework and taxonomy of team processes," *Academy of Management Review*, vol. 26, no. 3, pp. 356–376, 2001.
[6] R. Groß, S. Nouyan, M. Bonani, F. Mondada, and M. Dorigo, "Division of labour in self-organised groups," in *From Animals to Animats 10*, M. Asada, J. C. T. Hallam, J.-A. Meyer, and J. Tani, Eds. Springer, 2008, pp. 426–436.
[7] C. Bererton and P. K. Khosla, "Towards a team of robots with repair capabilities: A visual docking system," in *Experimental Robotics VII*, D. Rus and S. Singh, Eds. Springer, 2001, pp. 333–342.
[8] J. S. Jennings, G. Whelan, and W. F. Evans, "Cooperative search and rescue with a team of mobile robots," in *Proc. of the Int. Conf. on Advanced Robotics*, 1997, pp. 193–200.
[9] L. Esterle and P. R. Lewis, "Online multi-object k-coverage with mobile smart cameras," in *Proc. of the Int. Conf. on Distributed Smart Cameras*. ACM, 2017, pp. 107–112.
[10] L. Esterle, "CHAINMAIL: Distributed coordination for multi-task k-assignment using autonomous mobile IoT devices," in *Proc. of the Int. Conf. on Distributed Computation in Sensor Systems*. IEEE, 2018, pp. 1–8, accepted for publication.
[11] G. Kerth, *Group decision-making in animal societies*. Springer, 2010, pp. 241–265.
[12] L. Conradt and C. List, "Group decisions in humans and animals: a survey," *Philosophical Trans. of the Royal Society of London B: Biological Sciences*, vol. 364, no. 1518, pp. 719–742, 2009.
[13] S. W. Pacala, D. M. Gordon, and H. C. J. Godfray, "Effects of social group size on information transfer and task allocation," *Evolutionary Ecology*, vol. 10, no. 2, pp. 127–165, 1996.
[14] G. Theraulaz, E. Bonabeau, and J.-L. Deneubourg, "Response threshold reinforcements and division of labour in insect societies," *Proc. of the Royal Society B: Biological Sciences*, vol. 265, no. 1393, pp. 327–332, 1998.
[15] S. Zaccaro, M. Marks, and L. DeChurch, Eds., *Multiteam systems*. Routledge, New York, 2012.
[16] M. J. B. Krieger, J.-B. Billeter, and L. Keller, "Ant-like task allocation and recruitment in cooperative robots," *Nature*, vol. 406, pp. 992 – 995, 2000.
[17] R. Grabowski, L. E. Navarro-Serment, C. J. Paredis, and P. K. Khosla, "Heterogeneous teams of modular robots for mapping and exploration," *Autonomous Robots*, vol. 8, no. 3, pp. 293–308, 2000.
[18] L. E. Parker and B. A. Emmons, "Cooperative multi-robot observation of multiple moving targets," in *Proc. of the Int. Conf. on Robotics and Automation*, vol. 3, 1997, pp. 2082–2089 vol.3.
[19] B. B. Werger and M. J. Matarić, "From insect to internet: Situated control for networked robot teams," *Annals of Mathematics and Artificial Intelligence*, vol. 31, no. 1, pp. 173–197, 2001.
[20] B. Jung and G. S. Sukhatme, "Cooperative multi-robot target tracking," in *Distributed Autonomous Robotic Systems 7*, 2006, pp. 81–90.
[21] A. Kolling and S. Carpin, "Cooperative observation of multiple moving targets: an algorithm and its formalization," *The International Journal of Robotics Research*, vol. 26, no. 9, pp. 935–953, 2007.
[22] C.-F. Huang and Y.-C. Tseng, "The coverage problem in a wireless sensor network," *Mobile Networks and Applications*, vol. 10, no. 4, pp. 519–528, 2005.
[23] M. Hefeeda and M. Bagheri, "Randomized k-coverage algorithms for dense sensor networks," in *Proc. of the Int. Conf. on Computer Communications*, 2007, pp. 2376–2380.
[24] M. Elhoseny, A. Tharwat, X. Yuan, and A. E. Hassanien, "Optimizing k-coverage of mobile wsns," *Expert Systems with Applications*, vol. 92, pp. 142 – 153, 2018.
[25] B. Liu, O. Dousse, P. Nain, and D. Towsley, "Dynamic coverage of mobile sensor networks," *Trans. on Parallel and Distributed Systems*, vol. 24, no. 2, pp. 301–311, 2013.
[26] G. Fusco and H. Gupta, "Selection and orientation of directional sensors for coverage maximization," in *Proc. of the Int. Conf. on Sensor, Mesh and Ad Hoc Communications and Networks*, 2009, pp. 1–9.
[27] C. Micheloni, B. Rinner, and G. L. Foresti, "Video analysis in pan-tilt-zoom camera networks," *Signal Processing Magazine*, vol. 27, no. 5, pp. 78–90, 2010.
[28] J. C. SanMiguel, C. Micheloni, K. Shoop, G. L. Foresti, and A. Cavallaro, "Self-reconfigurable smart camera networks," *Computer*, vol. 47, no. 5, pp. 67–73, 2014.
[29] C. Piciarelli, L. Esterle, A. Khan, B. Rinner, and G. L. Foresti, "Dynamic reconfiguration in camera networks: A short survey," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 26, no. 5, pp. 965–977, 2016.
[30] A. Khan, B. Rinner, and A. Cavallaro, "Cooperative robots to observe moving targets: Review," *IEEE Trans. on Cybernetics*, vol. 48, no. 1, pp. 187–198, 2018.
[31] L. Esterle and B. Rinner, "An Architecture for Self-Aware IoT Applications," in *Proc. of the Int. Conf. on Accoustics, Speech, and Signal Processing*. IEEE, 2018, pp. 1–5, accepted for publication.
[32] L. Esterle, P. R. Lewis, H. Caine, X. Yao, and B. Rinner, "Camsim: A distributed smart camera network simulator," in *Proc. of the Int. Conf. on Self-Adaptive and Self-Organizing Systems Workshops*. IEEE, 2013, pp. 19–20.
[33] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998.
[34] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine Learning*, vol. 47, no. 2, pp. 235–256, 2002.
[35] R. Grosu, D. Peled, C. R. Ramakrishnan, S. A. Smolka, S. D. Stoller, and J. Yang, "Using statistical model checking for measuring systems," in *Leveraging Applications of Formal Methods, Verification and Validation. Specialized Techniques and Applications*, T. Margaria and B. Steffen, Eds. Springer, 2014, pp. 223–238.
[36] A. Lukina, L. Esterle, C. Hirsch, E. Bartocci, J. Yang, A. Tiwari, S. A. Smolka, and R. Grosu, "Ares: Adaptive receding-horizon synthesis of optimal plans," in *Tools and Algorithms for the Construction and Analysis of Systems*, A. Legay and T. Margaria, Eds. Springer, 2017, pp. 286–302.
[37] T. Hérault, R. Lassaigne, F. Magniette, and S. Peyronnet, "Approximate probabilistic model checking," in *Proc. of the Int. Conf. on Verification, Model Checking, and Abstract Interpretation*, 2004.