

# CHAINMAIL: Distributed coordination for multi-task $k$ -assignment using autonomous mobile IoT devices

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

**Abstract**—The Internet-of-Things (IoT) becomes more and more pervasive and supports us in our daily activities. However, when individual devices struggle in accomplishing certain tasks, they have to cooperate in order to achieve desired outcomes. In the absence of a central controller, devices have to coordinate autonomously within the network in order to attain and complete as many tasks as possible. We propose CHAINMAIL, a novel, distributed approach to coordinate sensors to attain tasks that cannot be accomplished by single, but only by cooperation of multiple devices. We demonstrate our approach with an IoT case study on multi-object  $k$ -coverage with autonomously operating mobile cameras and show that our approach does not over-provision tasks, allowing the remaining devices to attain other duties. This enables our network to provision more tasks in the same time as other comparable solutions.

## I. INTRODUCTION

The Internet-of-Things (IoT) is comprised of a variety of sensors and actuators, interacting with each other and exchanging information to complete their various tasks. This allows us to get more contextual information or support us to achieve our goals altogether.

However, not in all cases tasks can be accomplished satisfactorily by a single device, requiring active and concurrent cooperation among multiple devices on a specific task. Furthermore, many of those networks are deployed *ad hoc* and consist of a large number of devices making a centralised approach infeasible. Having each individual device controlled by an autonomous software agent, enables them to coordinate themselves in a distributed fashion in order to achieve common tasks. Distributed coordination gets more pressing when devices are mobile and able to relocate during runtime. This makes the local environment subject to constant change which may not be accounted for *a priori*.

Coordinating mobile devices to attain tasks brings about various benefits but at the same time a number of challenges that need to be tackled. On one side, idle devices can help others to provision tasks and allow to complete common tasks quicker and with less resources than the individual. On the other side, devices need to make sure they do not miss tasks within the environment that may require their attention and, at the same time, do not over-provision tasks which may lead to wasting valuable resources.

Exploiting the above benefits and tackling the challenges simultaneously, is the main goal of this paper. More precisely,

we are interested in assigning  $k$  devices to changing set of unknown, moving tasks. This assignment should not over- or under-provision tasks and at the same time maximise the number of overall attained tasks. In order to achieve this in a distributed fashion, we propose CHAINMAIL, an approach to assign  $k$  devices to a changing set of moving tasks in the absence of a central control.

The problem was introduced by Esterle and Lewis [1] and is a combination of the  $k$ -coverage problem introduced by Huang and Tseng [2] and the Cooperative Multi-Robot Observation of Multiple Moving Targets problem (CMOMMT) [3] introduced by Parker and Emmons. Huang and Tseng optimise coverage of the monitored region by at least  $k$  sensors and use a single-shot optimisation technique to do so. However, when tasks move, a one time optimisation will not provide satisfactory outcomes, unless the entire region can be covered by  $k$  sensors to begin with. This gives rise to an online version of  $k$ -coverage in order to attain as many tasks as possible with at least  $k$  devices. In contrast, CMOMMT tries to cover objects moving in the environment. However, their goal is not to cover them with multiple sensors but with at least one sensor at any time. Furthermore, the set of objects remains stable throughout the experiment, ensuring that each device is covering an object to continuously provide good performance. In our case, the set of tasks can change dynamically at runtime, due to newly arising and disappearing, or resolving, tasks. This renders attaining devices of resolved tasks useless to the network-wide performance evaluation. Therefore we derive the following research questions:

- 1) If devices are controlled by individual agents, making decisions based only on local information, can they coordinate themselves in a distributed fashion in order to provision tasks with  $k$  devices while keeping the number of tasks attained by more than  $k$  devices to a minimum? In other words, can we provision each task with exactly  $k$  devices?
- 2) Furthermore, can agents coordinate in order to maximise the number of attained tasks while at the same time maximising the number of tasks provisioned with at least  $k$  devices? In other words, can we maximise both, the number of provisioned tasks in general and the number of tasks provisioned with at least  $k$  devices?

To demonstrate the benefits of CHAINMAIL, we use a visual sensor network case study employing mobile smart cameras individually operated by software agents trying to observe objects moving around in the environment. In addition to collaborative sensing and search and track/rescue applications, other areas are collaborative construction [4], edge/fog computing [5], [6] or multi-robot interactions (e.g. collaborative drilling) [7].

In the following section of this paper a formal problem definition for  $k$ -assignment with respect to the corresponding case study is given. Section III discusses related work. Section IV presents CHAINMAIL for distributed coordination for multi-task  $k$ -assignment. Furthermore, we discuss the initiation problem of such an approach and how communication range impacts the performance of our approach to assign exactly  $k$  devices to each task. Section V presents the results with respect to maximise the number of attained tasks while increasing the number of tasks provisioned by  $k$  devices. We conclude our paper with an outlook on potential future work in Section VI.

## II. PROBLEM STATEMENT

The multi-task  $k$ -assignment problem can be stated as follows.

Consider a set of tasks  $O_t = \{o_1, o_2, \dots, o_m\}$  at time  $t$  that need to be provisioned and a set of mobile devices  $D = \{d_1, d_2, \dots, d_n\}$ . Importantly, the set of devices remains constant while the set of tasks may change over time as new tasks arise and others disappear. Tasks are unknown to the devices until a task is within the sensing range of a device. They arise and remain with a probability of  $\sigma$  and a duration of  $\gamma$ , respectively. Both, tasks and devices, can move around in a 2D plain with a velocity  $\vec{s}_i(t)$ . Their location at time  $t$  is denoted as  $\vec{x}_i(t) = (x_i, y_i)$ . Specifically, the discrete-time behaviour of a device  $d_i$  can be defined as

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{s}_i(t) \quad (1)$$

This velocity  $\vec{s}_i$  is controlled by an internal autonomous software agent.

Devices 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 device has a dedicated sensing/actuating area  $f_i$ . We consider a task  $o_a$  to be provisioned by a device  $d_i$  if the task is within the sensing/actuating area  $f_i$ . Therefore, a device's state is defined as  $d_i = \langle \vec{x}_i, \vec{s}_i, f_i, c_i \rangle$ . This defines a snapshot at a particular point in time and can be further indexed by  $t$  to represent the device's state over time.

The mobile devices 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 device  $d_i$ , if the task is geometrically within  $f_i$ :

$$prv(o_a, f_i, t) = \begin{cases} 1, & \text{if } \vec{x}_a(t) \text{ is inside } f_i \\ 0, & \text{otherwise,} \end{cases}$$

However, each task requires  $k$  devices to be accomplished. Therefore, the second goal requires them to maximise the

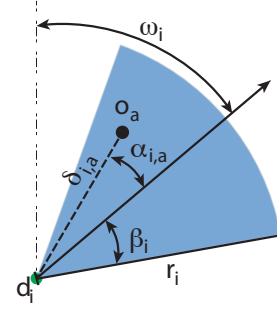


Figure 1. Illustration of a task in a camera's FoV. FoV is illustrated in blue with a range  $r_i$ , an orientation  $\omega_i$ , and angle  $\beta_i$  on both sides of  $\omega_i$ . The task is at angle  $\alpha_{i,a}$  to the camera's orientation and distance  $\delta_{i,a}$ .

number of task provisioned with  $k$  devices. We consider a task  $o_a$  to be provisioned by  $k$  devices at a given time  $t$  if

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

For a given value of  $k$ , provided by an operator and known to all devices, we are interested in the *performance* of the network. This is the number of tasks being provisioned by  $k$  devices over the entire duration  $T$  of our experiment:

$$performance = \frac{\sum_{t=1}^T \sum_{j=1}^m kprv(o_a, k, t)}{\max(prv)} \quad (2)$$

where  $\max(prv)$  is the maximum  $\sum_{t=1}^T \sum_{j=1}^m prv(o_a, t)$  across all tested approaches. This *performance* allows us to compare different approaches with each other.

For our specific case-study, using mobile smart cameras, we define the sensing/acting area  $f_i$  of device  $d_i$  as the respective field of view (FoV), which is modelled as a cone and defined by a range  $r_i$ , an angle  $\omega_i$ , defining the viewing orientation relative to a fixed reference point, and an angle  $\beta_i$ , defining the width on either side of  $\omega_i$ . The range of a camera is limited by the distance at which an object can be detected and identified using the on-board camera. To change the orientation of  $f_i$ , each camera  $d_i$  has a rotational speed  $u_i$ . We can define the discrete-time behaviour for the orientation as

$$\omega_i(t+1) = \omega_i(t) + u_i(t), \quad (3)$$

where  $u_i$  is again controlled by an autonomous internal software agent. The agent will adjust  $u_i$  and  $\vec{s}_i$  in such a way to keep a designated task  $o_a \in O_t$  within  $f_i$ . This is done independently of CHAINMAIL and illustrated for the case of an unidirectional visual sensor in Figure 1. Importantly, each device only has a single designated task it actively provisions and follows. However, other tasks within its sensing/actuating area are also provisioned. In a smart camera network, a task is an object that exhibits specific behaviour. This can be determined by the individual camera [8]. The object is removed from the set of tasks if it does not exhibit a given behaviour anymore.

## III. RELATED WORK

The problem tackled in this paper brings together the  $k$ -coverage problem introduced by Huang and Tseng [2] and the CMOMMT problem introduced by Parker and Emmons [3].

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 [9] propose a distributed approach to approximate optimal  $k$ -coverage in a network. Elhoseny et al. [10] 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. [11] analyse the benefits of moving sensors to detect and cover specific, but unknown, stationary points in the environment. Fusco and Gupta [12] propose a simple greedy algorithm to optimally place and orient directed sensor for  $k$ -coverage of static objects in the environment.

In contrast Cooperative Multi-robot Observation of Multiple Moving Targets (CMOMMT), tries to find and cover moving targets that are initially unknown. Werger and Mataric [13] 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 [14] 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 [15] perform a target loss prediction allowing the individual devices to call for help in a timely manner.

In camera networks, Micheloni et al. [16] 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 [17], [18], [19]. However, the problem of  $k$ -coverage with unknown number of targets using mobile devices only received little interest yet.

#### IV. CHAINMAIL

We propose CHAINMAIL, a distributed approach to provision tasks with  $k$  devices. The fundamental idea to CHAINMAIL is that devices detecting tasks notify others to provision it. In contrast to previous approaches, CHAINMAIL does not continuously notify  $k - 1$  other devices<sup>1</sup>, but notifies only another single device at random, which in turn, passes this message on until  $k$  devices agreed to provision the task. The CHAINMAIL algorithm is given in Algorithm 1.

Each transmitted message to notify another device contains information about the task  $o_a$ , its location  $\vec{x}_a$  within the environment, a list of previous recipients  $\rho$  of the notification message, and  $k$  indicating how many devices are still required to provision the task:  $m = \{o_a, \vec{x}_a, \rho, k\}$ .

Furthermore, each device  $d_i$  locally manages a “blacklist”  $\ell_i$  of devices which not to notify. Each device would update this local list with devices having recently sent notifications

<sup>1</sup>or  $k$  other devices, depending whether or not the detecting device is able to provision the task or not.

or being on a list of recipients  $\rho$  in a received notification message. Using  $\ell_i$  allows each individual device to locally make a profound decision whether to communicate with another device or not. However, devices only operate on local information and do not constantly exchange information about their state with other devices. We introduce a *forgetting parameter* in order to update the list over time and allow previously excluded devices to be notified again. Otherwise, each device would add all other devices in its list  $\ell$  over time and would not be able to notify any other device about detected tasks. We use a simple *forgetting parameter*  $\phi_j$  for each device  $d_j \in \ell$ . We initialise  $\phi_j = \tau$ . Over time,  $\phi_j$  will be reduced by a forgetting rate  $\Delta$ :

$$\phi_j(t + 1) = \phi_j(t) - \tau * \Delta$$

When  $\phi_j \leq 0$  we remove  $\phi_j$  from  $\ell_i$  and device  $d_j$  will be considered again as potential collaboration partner to provision tasks. Alternatively, the forgetting parameter could also be used as an inverse probability to decide whether or not to notify another device.

##### A. CHAINMAIL initiation policy

CHAINMAIL is initiated whenever a device detects a new task, of which it was not previously aware. However, this task might have been detected before by another device and the current device is not aware of this situation. Therefore we need to make sure only a single device initiates CHAINMAIL for a given task at any time.

In order to ensure this, each device will send information about its intention to start CHAINMAIL to all devices in its communication range. If another device has already requested others to provision this particular task, the device will be notified and will not initiate the CHAINMAIL again. We can assume that a device that has initiated CHAINMAIL will also stay close to the respective task and hence receive the announcement of intent of another device. However, this is not guaranteed as at least one device knowing about the initiated CHAINMAIL needs to be close enough to receive the initiation request from the other device. This might not be the case when either the CHAINMAIL message travelled in the opposite direction of the new device, or the communication range of each device equals only the sensing range and the other provisioning device is on the opposite side of the task and hence outside the communication range.

In case two devices detect the task at the same time, time stamps of the messages can be used to determine the device being allowed to initiate the CHAINMAIL. Additionally, the first notification of CHAINMAIL could be directed to the other device, ensuring that both already observing devices are also provision the task right away.

While this initiation policy does not overcome the *hidden terminal problem* [20], it allows to limit the number of over-provisioning devices. In our case-study, CamSim operates as discrete simulation environment, and therefore we rely on simple time steps and first-come-first serve practice.

---

**Algorithm 1:** CHAINMAIL algorithm for distributed  $k$ -assignment for networks of mobile sensors
 

---

```

1 foreach  $d_i \in D$  at time  $t$  do
2   foreach  $o_a \in O$  &  $cov(o_a, f_i, t)$  do
3     Ask every device within range  $c_i$  (cp.
4     Section IV-B) if provisioning task  $o_a$ .
5     if Not provisioned yet then
6       Update  $\rho \cup \{d_i\}$  in message
7       Notify single random other device  $d_j$  within
8       vicinity  $c_i$ , where  $d_j \notin \ell_i$ 
9       Adjust  $u_i$  and  $\vec{s}_i$  to provision task  $o_a$  (Eq. 3
10      and 1)
11     end
12   end
13 if Received notification then
14   Update  $\rho \cup \{d_i\}$  in message.
15   if  $d_i$  provisions another task then
16     Notify another device  $d_j$ , where  $d_j \notin \rho$  and
17      $d_j \notin \ell_i$ 
18     Update  $\ell_i \cup \{d_j\}$ 
19   else
20     Adjust  $u_i$  and  $\vec{s}_i$  to provision task  $o_a$  (Eq. 3
21     and 1)
22     Update  $k = k - 1$  in message
23     if  $k > 0$  then
24       Notify another random device  $d_j$  within
25       vicinity  $c_i$  where  $d_j \notin \ell_i$  and  $d_j \notin \rho$ 
26       Update  $\ell_i \cup \{d_j\}$ 
27     end
28   end
29   Update local list  $\ell_i = (\rho \setminus \ell_i) \cup \ell_i$ 
30 else
31   Search for tasks
32 end
33 end

```

---

### B. Communication range

Esterle and Lewis [1] propose different strategies to limit communication. However, CHAINMAIL uses simple message passing in combination with randomly communicating with another device in order to achieve provisioning of all tasks with at least  $k$  devices. Therefore, we do not need specific strategies to focus our communication efforts. However, we analyse different communication distances for our random communication approach. We consider four different ranges:

- 1) **Single range (SR):** uses the visual range of the device as a communication range. The device can communicate with other devices within this distance.
- 2) **Double range (DR):** assumes the communication range is double the distance of the visual range.
- 3) **Triple range (TR):** similar to *double range*, *triple range* communication can communicate with any device within three times the distance of the visual range.
- 4) **Full area (FA):** the device can communicate with any

device within the scenario independent of the distance between those two devices.

We use a case-study in mobile camera networks tasked to provision and observe objects moving around in the environment to test CHAINMAIL. We used CamSim simulation environment [21] and the same experimental scenarios as in Esterle and Lewis [1]. An overview of the starting positions of the cameras in each scenario is given in Figure 2. Table I presents an overview of all scenarios, the employed number of cameras and maximum number of tasks (objects). Furthermore, the mean sum of time steps in which tasks are present and the mean sum of time steps in which tasks are provisioned by at least one device using the outcomes of the best approach averaged over 30 repeated experiments. We can observe that in most cases, 70% of the tasks are provisioned. We did not test communication ranges shorter than the sensing range as it would increase the chances of encountering the *hidden terminal problem* [20].

In our scenarios, tasks and devices cannot leave the simulation environment, leading to a constant number of devices and tasks. Devices and tasks move in a random fashion by following a straight path until reaching the boundary of the simulation environment from where they bounce back in a random direction. All our experiments were repeated 30 times to account for randomness. Each experiment lasts 1000 time steps. While we use a fixed set of tasks in our experiments, devices do not know about the total number of available tasks at any time. However, in the real world, tasks may not be limited in numbers or duration. We simulate this by using  $\sigma$  and  $\gamma$ . In all cases  $\tau = 10$ ,  $\Delta = 0.1$ ,  $k = 3$ ,  $\sigma = 0.05$  and  $\gamma$  is randomly drawn from [5, 100] with a uniform distribution.

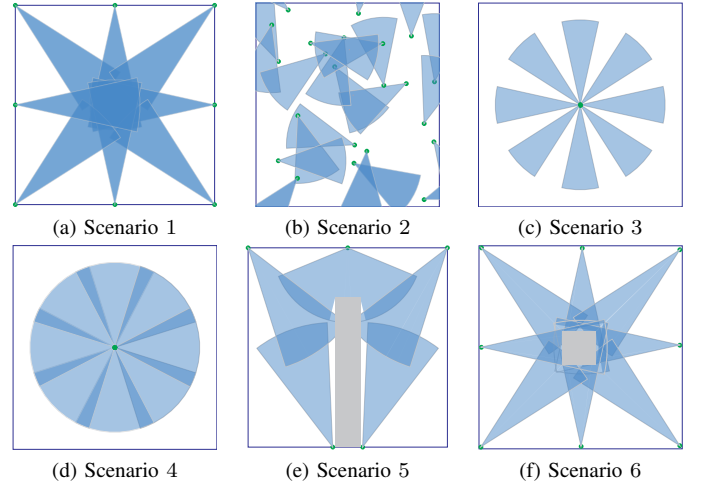


Figure 2. Evaluated case-study scenarios. Green dots represent cameras and blue cones their respective FoVs. Gray blocks illustrate opaque walls/areas.

Figure 3 shows a snapshot of the different number of cameras observing objects (provisioning tasks) for each of the 6 scenarios at time step 500—the middle of the experiment. Each bar represents the number of cameras provisioning known objects while the height of the bar shows the normalised mean number of objects of interest being provisioned. As with all

	Devices	Max. Conc. Tasks	Tasks Overall	Tasks Provisioned
Scenario 1	8	8	4503.4	3295.5 (73.2%)
Scenario 2	22	19	10680.7	8173.1 (76.5%)
Scenario 3	8	8	4493.7	2513.2 (55.9%)
Scenario 4	8	8	4515.1	2964.2 (65.7%)
Scenario 5	5	7	3976.2	3125.9 (78.6%)
Scenario 6	8	8	4514.0	3253.8 (72.0%)

Table I

OVERVIEW OF SCENARIOS. NUMBER OF DEVICES  $n$ , MAXIMUM NUMBER OF CONCURRENT TASKS  $m$  AT ANY DISCRETE TIME STEP, AVERAGE NUMBER OF TASKS SUMMED OVER ALL TIME STEPS, AVERAGE NUMBER OF PROVISIONED TASKS, SUMMED OVER ALL TIME STEPS, AND CORRESPONDING PERCENTAGE OF OVERALL TASKS.

our experiments, we tasked our network to cover all objects of interest with  $k = 3$  devices. It becomes apparent that limiting communication to those devices in the vicinity allows for more focused  $k$ -coverage where in comparison, communication with a wider range results in more than  $k$  cameras provisioning an object. We speculate this is due to devices travelling a longer distance in order to provision a specific object will also provision other random objects on the way. In smaller scenarios where the communication range covers (almost) the entire scenarios, this becomes less important.

## V. RESULTS: $k$ -COVERAGE USING CHAINMAIL

Results shown in Figure 4 give an overview of the average amount of objects being provisioned by cameras over all time steps and 30 experiments using CHAINMAIL. The results have been normalised by the total number of detected objects. In most cases objects are mostly provisioned by  $k = 3$  cameras. One of the main benefits of CHAINMAIL is its ability to keep over-provisioning of objects to a minimum. It is also apparent that there are a high number of outliers for objects provisioned by 1 or 2 cameras. These outliers come about the ‘‘accidental’’ coverage of objects when cameras pass by objects or objects move through the FOV of a camera without being actively provisioned.

Furthermore, we want to draw a comparison between the the approaches previously proposed by Esterle and Lewis [1] and CHAINMAIL. They investigate different communication strategies in combination with various response models A communication strategy refers to the decision of a camera to notify another camera in order to provision an object while the response models represent a local decision mechanism on each camera whether or not to follow such an notification. Each camera would use its communication strategy in each time step to notify new cameras as long as an object/task is within its FOV. They compare their approaches towards three types of non-coordinating cameras:

- 1) *Static*: cameras are not able to move at all, constantly remaining in their initial position.
- 2) *Random*: cameras moving randomly but not following objects of interest intentionally.
- 3) *Random and following*: cameras move randomly but follow objects of interest upon detection.

Their coordination approaches uses the following communication strategies: Broadcasting (BC) communicates with all cameras in the environment,  $k$ -closest (CL) communicates only with  $(k - 1)$  closest cameras, and  $k$ -random (RA) notifies  $(k - 1)$  other cameras in the environment. Using SMOOTH (SM) or STEP (ST), cameras learn local neighbourhood relations over time and use this information as probability or as a threshold, respectively, to select  $(k - 1)$  cameras for communication.

Results in Figure 5 show the normalised mean coverage of objects throughout all experiments for the individual approaches. Since the movement of cameras in the environment is determined by the approach they are using, cameras encounter different number of objects of interests. If we only normalise the outcome of the approach by the number of objects encountered, the dominant strategy would be to ensure that each encountered object is covered by at least  $k$  cameras independent of the number of objects/tasks in the environment. However, we are interested in covering as many objects with at least  $k$  cameras or more. Therefore, we normalised the result of the individual approach by the maximum number of encountered interesting objects among all approaches. Interestingly, throughout all experiments, *static* cameras (not moving nor coordinating their provisioning) detected the highest number of objects. This is due to the initially good coverage of the different networks but also due to moving cameras starting to cluster around moving objects, leaving a majority of the remaining area uncovered. However, the specifics of this remains subject for future research.

The bars in Figure 5 show the mean coverage of objects normalised by the average maximum number of objects detected across all approaches and experiments for the respective scenario. The yellow (top) section of each bar indicates the normalised number of objects provisioned by one camera, the green (middle) section shows the amount of objects provisioned by two cameras, and the blue (lower) section represents the portion of objects provisioned by three or more cameras. The benefit of CHAINMAIL is apparent when trying to achieve  $k+$ -coverage. However, to cover objects with at least 1 camera, Random Following is achieving the best results for the first 4 scenarios. We speculate that the low performance in coverage for Scenario 5 and 6 is coming about the added opaque walls, representing an obstacle for objects and cameras, and allowing objects to suddenly disappear from the FOV of a camera. Throughout all experiments, CHAINMAIL has outperformed all coordinated approaches previously presented by Esterle and Lewis [1] in terms of achieving 1-coverage. In most cases, CHAINMAIL is even able to achieve higher  $k$ -coverage than previously proposed approaches.

## VI. CONCLUSION

We presented CHAINMAIL, a distributed approach to assign  $k$  devices to a set of unknown and moving tasks. The approach achieves a high proportion of attained tasks while not over-provisioning tasks at the same time. Furthermore, we

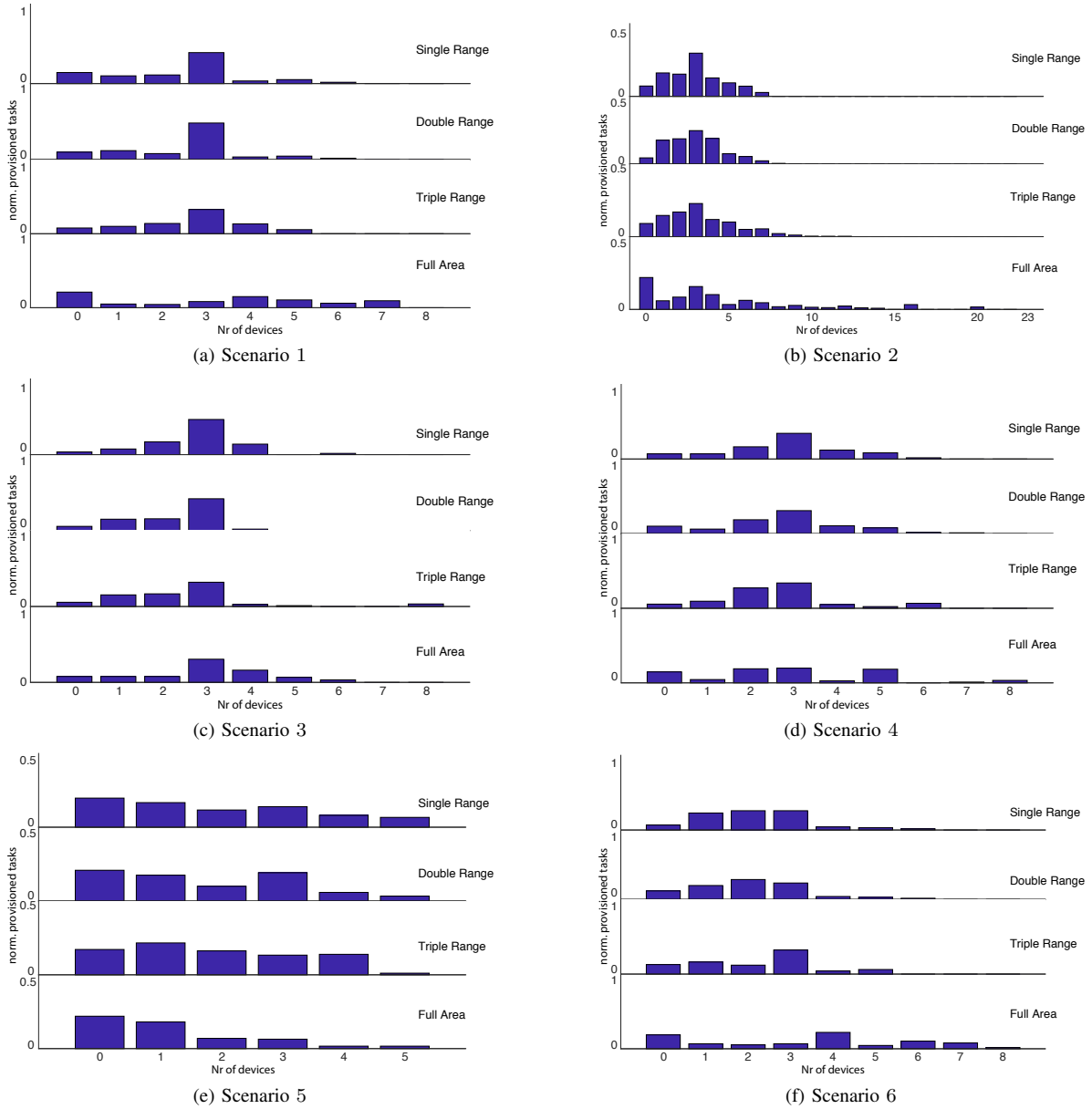


Figure 3. Comparison of provisioning of tasks using different communication distances by the devices in the network for 6 scenarios. Each bar represents the number of devices provisioning the percentage of tasks known to the network. Experiments have been performed 30 times and mean results are shown.

showed the effect of different ranges for communication on the achieved  $k$  coverage.

For future work, we first plan to deploy CHAINMAIL on real cameras to test its feasibility in real world scenarios. Second, for directed sensors, devices do not consider the orientation of already provisioning devices. Incorporating this information would allow cameras to provide new information when provisioning an object from a new angle. However, this would introduce a trade-off for each camera between (a) the effort to get to a position to provide information from a new angle and (b) the additional acquired information of doing so. Finally, we will explore the relationship between frequency of newly arising tasks, the density of devices and the size of the scenario in order to formulate (statistical) guarantees for

CHAINMAIL.

#### ACKNOWLEDGMENTS

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. The author would also like to thank Peter Lewis for the fruitful discussions.

#### REFERENCES

- [1] 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. 1–6.
- [2] 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.
- [3] L. E. Parker and B. A. Emmons, "Cooperative multi-robot observation of multiple moving targets," in *Int. Conf. on Robotics and Automation*, vol. 3, 1997, pp. 2082–2089 vol.3.



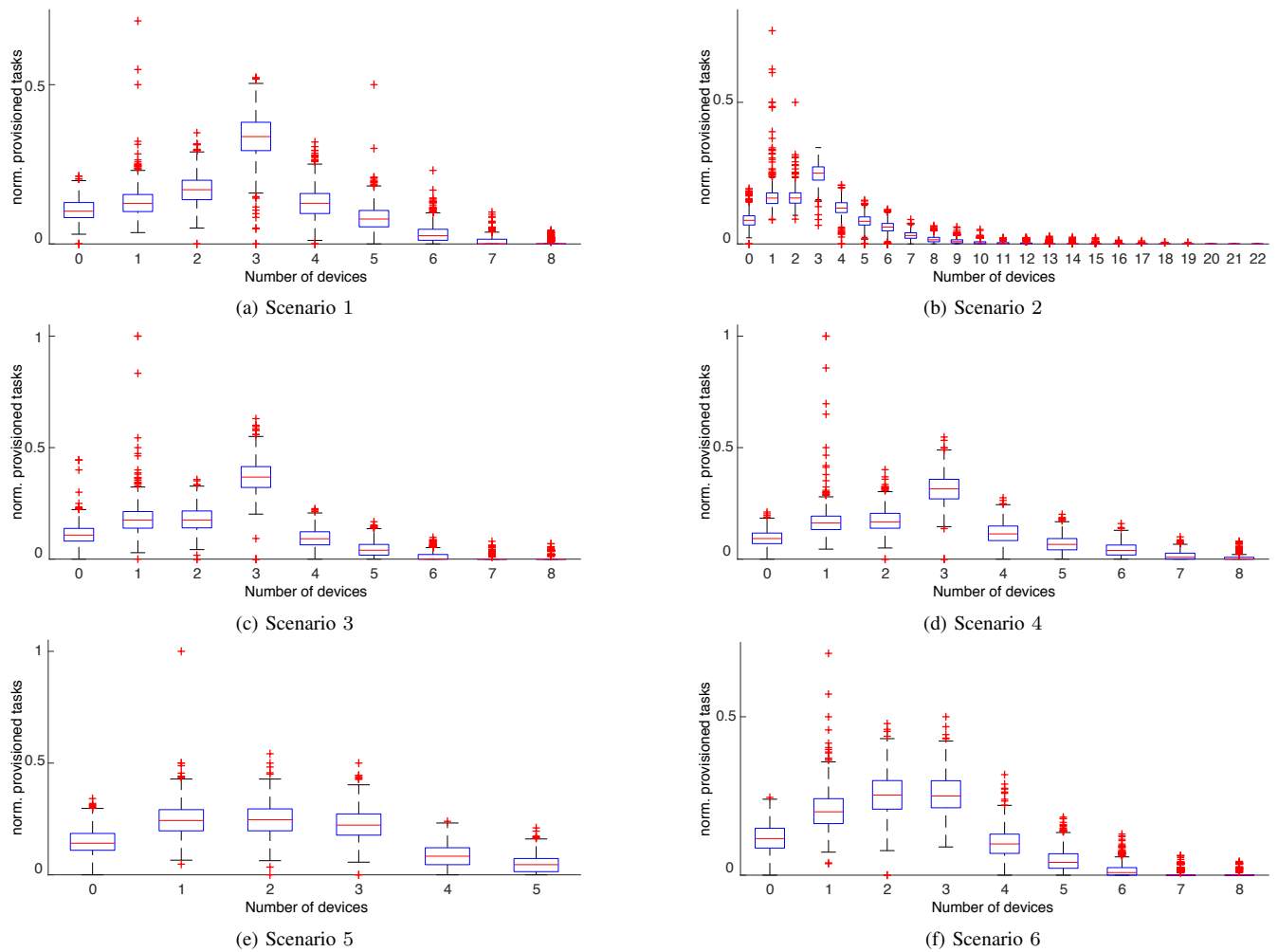
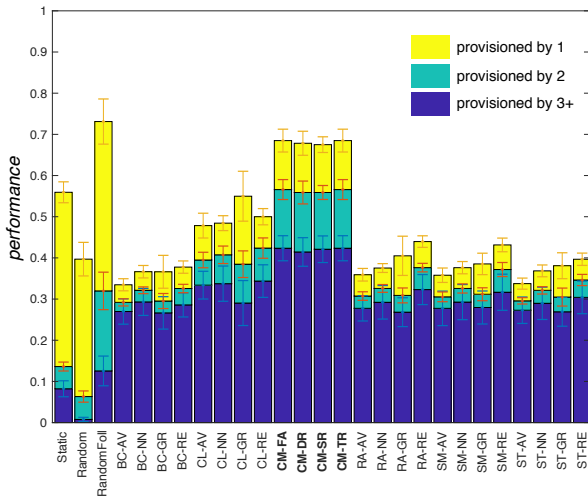
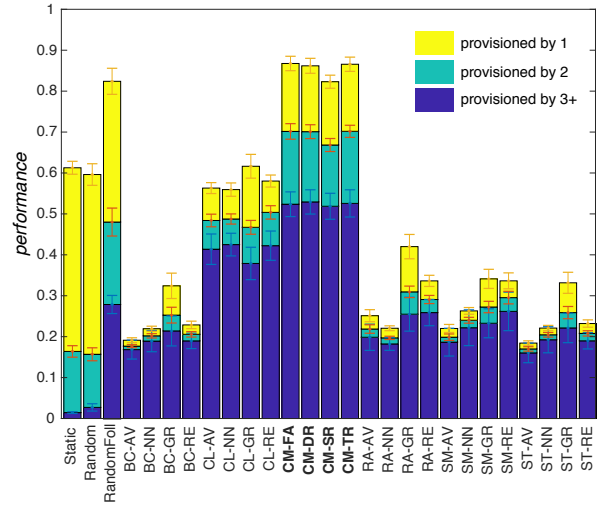


Figure 4. Box plots showing the median, first and third quartile of  $k$ -assignment for  $k = 3$  for our 6 scenarios. Outliers are indicated as red plus-signs (+). Devices communicate with single range (SR).

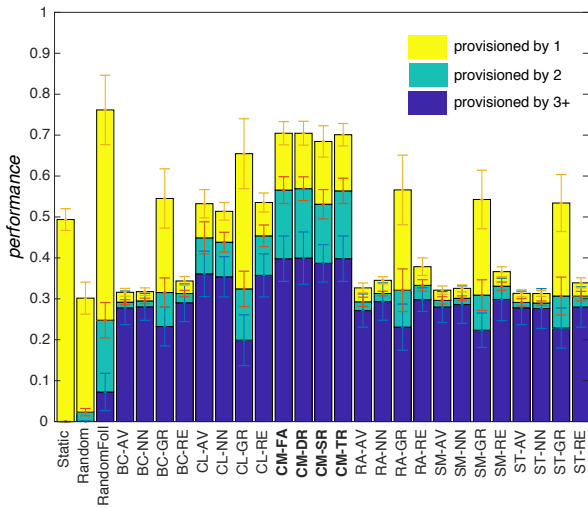
- [4] F. Augugliaro, S. Lupashin, M. Hamer, C. Male, M. Hehn, M. W. Mueller, J. S. Willmann, F. Gramazio, M. Kohler, and R. D'Andrea, "The flight assembled architecture installation: Cooperative construction with flying machines," *Control Systems*, vol. 34, no. 4, pp. 46–64, 2014.
- [5] W. Shi and S. Dustdar, "The promise of edge computing," *Computer*, vol. 49, no. 5, pp. 78–81, 2016.
- [6] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.
- [7] B. Glass, H. Cannon, C. Stoker, and K. Davis, *Robotic and Human-tended Collaborative Drilling Automation for Surface Exploration*. American Institute of Aeronautics and Astronautics, 2006.
- [8] T. Ko, "A survey on behavior analysis in video surveillance for homeland security applications," in *Applied Imagery Pattern Recognition Workshop*, 2008, pp. 1–8.
- [9] M. Hefeeda and M. Bagheri, "Randomized  $k$ -coverage algorithms for dense sensor networks," in *Int. Conf. on Computer Communications*, 2007, pp. 2376–2380.
- [10] M. Elhoseny, A. Tharwat, X. Yuan, and A. E. Hassanien, "Optimizing  $k$ -coverage of mobile wsn," *Expert Systems with Applications*, vol. 92, pp. 142 – 153, 2018.
- [11] 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.
- [12] G. Fusco and H. Gupta, "Selection and orientation of directional sensors for coverage maximization," in *Communications Society Conf. on Sensor, Mesh and Ad Hoc Communications and Networks*, 2009, pp. 1–9.
- [13] B. B. Werger and M. J. Matarić, "From insect to internet: Situated control for networked robot teams," *Annals of Math. and A. I.*, vol. 31, no. 1, pp. 173–197, 2001.
- [14] B. Jung and G. S. Sukhatme, "Cooperative multi-robot target tracking," in *Distributed Autonomous Robotic Systems 7*, 2006, pp. 81–90.
- [15] A. Kolling and S. Carpin, "Cooperative observation of multiple moving targets: an algorithm and its formalization," *The Int. J. of Robotics Res.*, vol. 26, no. 9, pp. 935–953, 2007.
- [16] 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.
- [17] 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.
- [18] C. Piciarelli, L. Esterle, A. Khan, B. Rinner, and G. L. Foresti, "Dynamic reconfiguration in camera networks: A short survey," *Trans. on Circuits and Systems for Video Technology*, vol. 26, no. 5, pp. 965–977, 2016.
- [19] A. Khan, B. Rinner, and A. Cavallaro, "Cooperative robots to observe moving targets: Review," *Trans. on Cybernetics*, vol. PP, no. 99, pp. 1–12, 2016.
- [20] F. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part ii - the hidden terminal problem in carrier sense multiple-access and the busy-tone solution," *Trans. on Communications*, vol. 23, no. 12, pp. 1417–1433, 1975.
- [21] L. Esterle, P. R. Lewis, H. Caine, X. Yao, and B. Rinner, "Camsim: A distributed smart camera network simulator," in *Int. Conf. on Self-Adaptive and Self-Organizing Systems Workshops*, 2013, pp. 19–20.



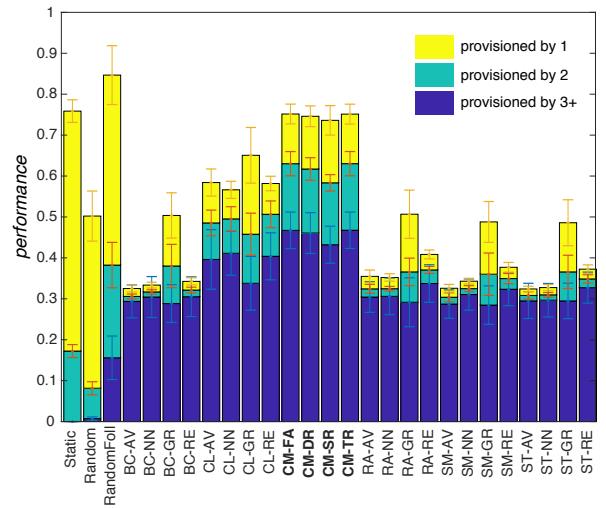
(a) Scenario 1



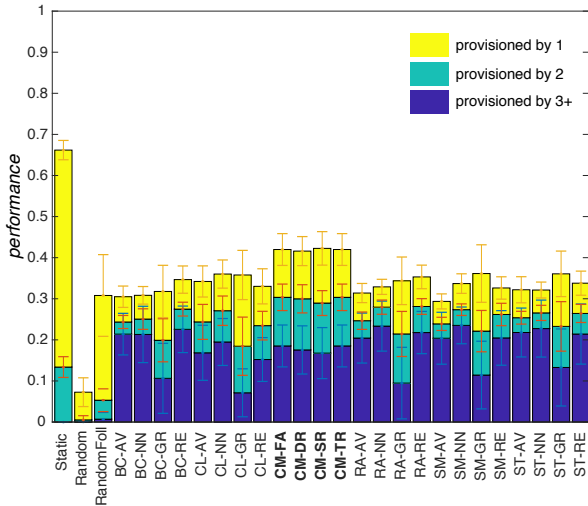
(b) Scenario 2



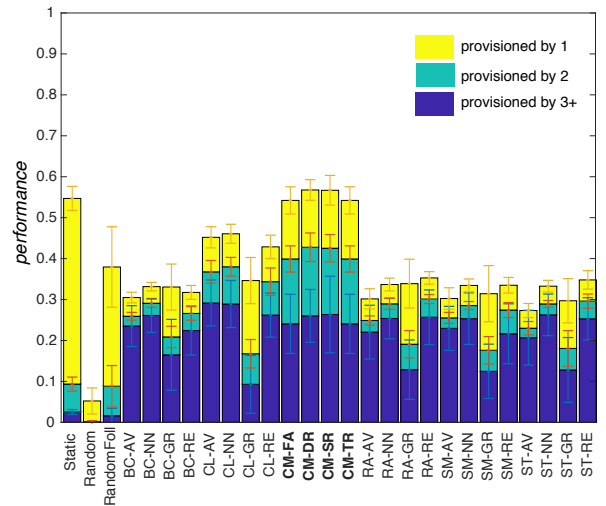
(c) Scenario 3



(d) Scenario 4



(e) Scenario 5



(f) Scenario 6

Figure 5. Mean provision of tasks (observing objects) normalised by the average maximum number of tasks detected across all approaches and experiments for the respective scenario as defined in Equation 2. The yellow (top) section of each bar indicates the normalised number of tasks provisioned by one device, the green (middle) section shows the amount of tasks provisioned by two devices, and the blue (lower) section represents the portion of tasks provisioned by three or more devices. BC corresponds to Broadcast communication, CL to communication with closest neighbours and RA with random neighbours. SM and ST exploit learnt neighbourhood relationships. AV corresponds to available response model, NN to Nearest-Newest, RE to received requests and GR to graph. FA is for Full Area, SR for Single Range, DR for Double Range and TR for Triple Range.