Static, Dynamic and Adaptive Heterogeneity in Distributed Smart Camera Networks

PETER R. LEWIS, Aston University
LUKAS ESTERLE, Alpen-Adria-Universität Klagenfurt and Lakeside Labs
ARJUN CHANDRA, University of Oslo
BERNHARD RINNER, Alpen-Adria-Universität Klagenfurt and Lakeside Labs
JIM TORRESEN, University of Oslo
XIN YAO, University of Birmingham

We study heterogeneity among nodes in self-organising smart camera networks, which use strategies based on social and economic knowledge to target communication activity efficiently. We compare homogeneous configurations, when cameras use the same strategy, with heterogeneous configurations, when cameras use different strategies. Our first contribution is to establish that static heterogeneity leads to new outcomes which are more efficient than those possible with homogeneity. Next, two forms of dynamic heterogeneity are investigated: non-adaptive mixed strategies, and adaptive strategies which learn online. Our second contribution is to show that mixed strategies offer Pareto efficiency consistently comparable with the most efficient static heterogeneous configurations. Since the particular configuration required for high Pareto efficiency in a scenario will not be known in advance, our third contribution is to show how decentralised online learning can lead to more efficient outcomes than the homogeneous case. In some cases, outcomes from online learning were more efficient than all other evaluated configuration types. Our fourth contribution is to show that online learning typically leads to outcomes more evenly spread over the objective space. Our results provide insight into the relationship between static, dynamic and adaptive heterogeneity, suggesting that all have a key role in achieving efficient self-organisation.

Categories and Subject Descriptors: C.2.3. [**Computer-Communication Networks**]: Network Management; C.2.4. [**Distributed Systems**]: Distributed applications; I.2.11 [**Distributed Artificial Intelligence**]: Coherence and coordination; I.4.9. [**Applications**]

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: heterogeneity; variation; learning; self-organisation; distributed smart cameras

ACM Reference Format:
Peter R. Lewis, Lukas Esterle, Arjun Chandra, Bernhard Rinner, Jim Torresen and Xin Yao, 2014. Static, Dynamic and Adaptive Heterogeneity in Distributed Smart Camera Networks. *ACM Trans. Autonom. Adapt. Syst.* 0, 0, Article 0 (2014), 30 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

The research leading to these results was conducted in the EPiCS project and received funding from the European Union Seventh Framework Programme under grant agreement n° 257906. http://www.epics-project.eu/

Authors' addresses: P. R. Lewis, Aston Lab for Intelligent Collectives Engineering (ALICE), Systems Analytics Research Institute (SARI), School of Engineering and Applied Science, Aston University, UK; X. Yao, CERCIA, School of Computer Science, University of Birmingham, UK; L. Esterle and B. Rinner, Institute of Networked and Embedded Systems, Alpen-Adria-Universität Klagenfurt and Lakeside Labs, Austria; A. Chandra and J. Torresen, Department of Informatics, University of Oslo, Norway.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© 2014 ACM 1556-4665/2014/-ART0 $15.00
DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

Collective systems are often designed such that constituent nodes are each given a local objective to pursue, and the system wide behaviour is a product of the actions and interactions of the nodes [Di Marzo Serugendo et al. 2011; Wooldridge 2001]. In pursuing local objectives, nodes are typically endowed with a common algorithm or behavioural strategy. However, nodes are often located in different areas, having different views of the world, and are subject to different experiences. In these cases, adopting different algorithms from each other may enable them to better achieve their own local objectives. It has also been shown that such heterogeneity among nodes can lead to better achievement of system wide objectives [Campbell et al. 2011; Anders et al. 2012], especially when nodes can adapt independently in response to uncertainty and changes in the environment during the system's lifetime [Salazar et al. 2010].

In this paper we study the effect of heterogeneity among nodes in a distributed smart camera network. Smart cameras are fully computationally capable devices endowed with a visual sensor, and typically run computer vision algorithms to analyse captured images. While standard cameras can only provide plain images and videos, smart cameras can pre-process these videos and provide users with aggregated data and logical information. In surveillance applications, smart cameras are used to provide an operator with data such as location, speed, and direction of an object, which could for example be a vehicle, person, ball etc. This is referred to as object tracking. Since smart cameras are designed to have a low energy footprint, their processing capabilities are also low. Therefore, typically each object of interest is tracked by only one camera at a time. Communication between cameras allows the network as a whole to track objects in a distributed fashion, handing over object tracking responsibilities from camera to camera as objects move through the environment. Previous work [Esterle et al. 2014] showed that by endowing cameras with self-interested agents, which traded responsibilities for tracking objects in an artificial market, the network as a whole could achieve an efficient allocation of objects to cameras, without any central coordination or *a priori* knowledge of the network topology. The cameras use pheromone-based online learning to determine which other cameras they trade with most often. This leads to a local neighbourhood relationship graph, also called the *vision graph*. This learnt vision graph represents adjacencies between cameras' fields of view, and enables them to selectively target their auction invitations, achieving high levels of tracking performance, while reducing communication and processing overhead.

Six different behavioural strategies were used by camera nodes, which determined the level of marketing activity they undertook, given the learnt vision graph. Some strategies incurred higher overheads but typically obtained higher levels of tracking performance; other strategies obtained the opposite results. However, the trade-off realised by each strategy was found to be highly scenario dependent; as camera positions varied and object movements differed, the relative benefits of the strategies were greatly influenced. Additionally, cameras often operated inefficiently since the homogeneous deployment of strategies forces a *one size fits all* approach, despite local differences in the vicinities of the cameras. As we have preliminarily demonstrated [Lewis et al. 2013], in this paper we show further that by permitting heterogeneity between cameras in terms of their strategies more Pareto efficient global outcomes can be obtained. In addition, in this paper we show that restricting individual cameras to a single strategy for their entire lifetime can also be inefficient. By endowing cameras with mixed strategies, where they select a strategy randomly at each decision point according to a fixed probability distribution, further Pareto efficiency can sometimes be obtained, relative to the static heterogeneous case.

Although heterogeneity can improve global efficiency, given the virtually limitless possibilities for camera network deployments and accompanying environmental dynamics, identifying by hand the most appropriate configuration at a particular point in time is not feasible. To overcome this problem we propose using online learning algorithms, specifically multi-armed bandit problem solvers (e.g. [Auer et al. 2002]) within each camera to learn the appropriate strategy for each node during runtime. These so-called bandit solvers balance exploitation behaviour, where a camera achieves high performance by using its currently known best strategy, with exploration, where the camera explores the effect of using other strategies to build up its knowledge. By employing bandit solvers in each camera, we are able to obtain global outcomes which are comparable with the exhaustively calculated Pareto efficient frontier arising from static heterogeneity. In some cases, the adaptive nature of the online learning algorithms extends the Pareto efficient frontier arising from the best static heterogeneous configurations. In many more cases, online learning algorithms extend the Pareto efficient frontier arising from the best mixed strategy configurations. We also find that, typically, outcomes arising from online learning are more evenly spread across the bi-objective space than those arising from a broad sample of mixed strategies. This is due to their ability to adapt to feedback during run time, and enables greater flexibility for an operator wishing to select an outcome reflecting their preference between the considered objectives. These results highlight an important role for heterogeneity in general, and for adaptive heterogeneity in particular, in the design and deployment of decentralised computational systems such as distributed smart camera networks.

The rest of the paper is structured as follows. In Section 2 we summarise recent work investigating heterogeneity and inter-agent variation in self-organising systems. We then provide a background to distributed smart camera networks and the object handover problem, and discuss the state of the art in this area. In Section 3 we formally introduce the problem studied and detail relevant aspects of the smart camera case study. In Section 4 we show how network level heterogeneity can improve global system performance, by analysing the effect of static pre-determined heterogeneity. In Section 5 we extend our analysis to the case where cameras employ mixed strategies, based on stationary probability distributions and show that these perform well, occasionally improving Pareto efficiency further. In Section 6 the online learning approach is introduced and evaluated. While the previous sections introduce the various forms of heterogeneity using visual representations of results from the simulation environment, Section 7 presents results from a real camera network deployment and Section 8 presents full quantitative results over all presented scenarios, evaluated for statistical significance. We conclude the paper and discuss future work in Section 9.

2. RELATED WORK

In this section, we firstly present and discuss recent advances in the understanding of the role of heterogeneity and variation in self-organising systems, with a particular focus on multi-agent software systems. Secondly, we provide a background to the case study used to investigate heterogeneity in this paper: the object handover problem in distributed smart camera networks. We survey the state of the art in approaches to tackling this problem, and describe the recent socio-economic handover approach used as a basis for work in this article.

2.1. Heterogeneity and Variation in Self-organising Systems

Nature provides numerous examples of heterogeneity (or variation or diversity) enabling populations to successfully self-organise to achieve their objectives [Campbell et al. 2011]. When using self-organisation to engineer decentralised collective systems, differences between system components can also be an important factor in enabling the

collective to obtain high performance [Prasath et al. 2009; Campbell et al. 2011]. Heterogeneity in sensor networks may take on various forms. Some of those which may be imagined include variation of hardware between nodes, differences in behaviour, and diverse parameters or objectives. In engineering such systems, the challenge is to find self-organisation algorithms which give rise to optimal forms of such heterogeneity, which in turn lead to high performance at the global level. Prasath et al. [2009] highlight two key issues:

(1) Whether heterogeneity allows optimisation beyond that possible in the homogeneous case, and
(2) What algorithms to use to achieve near-optimal heterogeneous networks.

Campbell et al. [2011] investigated the effect of inter-agent variation on a multi-agent task allocation problem, showing that such variation creates more possible *organisations* (configurations) of the system. This larger configuration space provides more possibilities, some of which may enable a collective system to better achieve its goal. The heterogeneity considered by Prasath et al. [2009] is in terms of the out-degree and wireless communication radius of nodes. They permit only two possibilities for each node's configuration, and compare the effect of using three different cooperative algorithms for determining node types, benchmarking the outcomes against ideal best possible outcomes. Rojković et al. [2012] present a technique for assigning roles to different nodes in a sensor network, which is compared with the near-optimal solution found by a genetic algorithm with global knowledge. Nakamura et al. [2009] reactively assign roles for data routing to different sensor nodes based on events to save energy during idle periods. Römer et al. [2004] propose the adaptation of nodes' roles based on their location and purpose. This adaptation is done using a predefined set of rules which are the same for all nodes in the network. In smart camera networks, Dieber et al. [2011] adapt the number of cameras in the network, changing their settings and the tasks being assigned to the cameras. They use a combination of an expectation-maximisation algorithm and evolutionary algorithm to satisfy predefined constraints. Finally, Nebehay et al. [2013] study the role of variation not between camera nodes, but as a characteristic of components within the object tracker in a single smart camera.

Salazar et al. [2010] highlight the importance of dynamic heterogeneous configurations for sensor networks deployed in uncharted environments, i.e. in scenarios about which there is a lack of *a priori* information. They argue that, in response to environmental changes over time, nodes should be able to reconfigure themselves according to local events, possibly in different ways from each other. Anders et al. [2012] also study the effect of inter-agent variation on the performance of a self-organising system in an uncertain environment. They found that in two algorithms, one based on schooling fish and the other on honey bees, the performance of the algorithms obtained a higher performance with heterogeneity. Their results suggest the presence of a critical threshold, a particular amount of variation required to ensure near-optimal solutions. They also found that in some cases too much variety could lead to negative effects such as oscillatory behaviour or slower arrival at the solution.

2.2. Distributed tracking and socio-economic handover

In this paper we study the role of heterogeneity in self-organising smart camera networks. These systems are sensor networks in which various computer vision tasks, such as object tracking, can be distributed among a group of cameras. When making the transition from object tracking in a single camera to an entire network of cameras, the responsibility for tracking has to be handed over between the cameras as the object moves. This *handover* process has to ensure that the next camera keeps track of each object as it moves between fields of view [Erdem and Sclaroff 2005]. To overcome the

handover problem, various approaches have been proposed, all of them varying in the assumptions made for the camera network, the available resources and the possibilities in distributing data and processing [Li and Bhanu 2009].

The main three assumptions found in prior work are (i) *a priori* knowledge about the scenario, (ii) central coordination, and (iii) a requirement for overlapping fields of view. Most early approaches such as presented by Quaritsch et al. [2007] or by Möller et al. [2008] facilitate pre-defined specific regions in the field of view (FOV) of each camera where the handover takes place. Other approaches presented by Makris et al. [2004], Detmold et al. [2007], and Javed et al. [2003] employ central components, collecting information from all cameras to determine how to allocate the tracking of objects of interest. Those not requiring a central component often rely on object correspondence in overlapping cameras (e.g. [Cheng et al. 2007; Mandel et al. 2007; Morioka et al. 2010]). While none of these approaches selects the optimal camera for tracking among all cameras in the network, the approaches by Li and Bhanu [2011], and Qureshi and Terzopoulos [2008] assign tracking responsibilities to the optimal camera based on user-defined objectives. Nevertheless, they require central coordination and rely on overlapping fields of view, respectively.

Only recently Esterle et al. [Esterle et al. 2014] presented a novel approach which removes all three assumptions. Their approach enables an autonomous and allocatively efficient assignment of object tracking responsibilities to cameras over time, without the need for *a priori* scenario knowledge or calibration. A decentralised market mechanism is used (Vickrey auctions are proposed) to determine the allocation of objects to cameras, and social knowledge associated with trading is learnt online using artificial pheromones. At the same time, this social knowledge is used to better target the cameras' marketing effort, and hence improve the efficiency of the entire system. We use this approach in this article, as a domain in which to study the space of heterogeneous and dynamic behaviour of a collective. Therefore, we further elaborate on the approach in this section.

Since the computational resources of smart cameras in a network are limited, typically only a single camera is responsible for tracking each object at a given time. This also applies when multiple cameras "see" the object at the same time. Cameras could simply track any object within their FOV. In cases with cameras having overlapping FOVs, this would result in objects being tracked by two cameras simultaneously and therefore in wasted resources from a network-wide perspective. Thus, the network has to coordinate the tracking responsibility for a given set of objects among the available cameras. When a camera has the responsibility for tracking an object, it is said to "own" that object. When a camera owns an object, the owning camera may also sell it to another camera, which corresponds to the handing over of tracking responsibilities from camera to camera. In this model, "selling" an object implies handing over responsibility for tracking it to another camera. Selling is determined by the outcome of a Vickrey auction, hosted by the selling camera, where cameras that can see the auctioned object place bids at a level equal to a utility value associated with that object by the camera. This utility is in turn equal to a chosen measure of the confidence or ability of the camera to track the object in question, given its image data.

In Esterle et al's simulation study, utility is the inverse of the Euclidean distance between the object and the camera. In their real camera network scenarios they use a visual tracking algorithm to determine the correlation between a defined model of the object of interest and the object within the FOV of the camera. This returns a *confidence* value, which is interpreted as utility. We adopt the same metrics in this paper. However, while the approach presented by Esterle et al. relies on a measurement of tracking quality, it is not important exactly how this is calculated, as long as it is equally defined for all participating cameras and confers a level of confidence

of having correctly identified the object. In both cases Esterle et al. assume perfect re-identification and tracking capabilities as well as lossless communication of the employed cameras.

A successful trade between cameras indicates that an object has moved from the field of view (FOV) of one camera to another. The amount of utility bid in the auction is transferred from the buyer to the seller. By observing the trading behaviour, the cameras learn, at runtime, a *vision graph* describing the spatial relationships between their FOVs. By using the learnt vision graph to inform their communication behaviour, Esterle et al. showed that cameras are able to reduce their communication overhead without significantly sacrificing tracking performance. Nevertheless, depending on how the vision graph is being exploited for advertising objects within the network but also how object move around in the environment, situations may occur where an object is not tracked by any camera even though it is visible to at least one camera. However this will be inherant to any approach which uses online learning. Further details of the auction-based handover mechanism are presented by Esterle et al. [Esterle et al. 2014], and are not directly relevant to the research questions studied in this paper. The crucial aspect of this approach to this paper is the choice of communication behaviour employed by cameras, which determines how objects are advertised, based on social information learnt at run time. It is therefore this behaviour, and its effect, that we will focus on for the remainder of this article.

3. PROBLEM STATEMENT

In this article, we are concerned with questions on the role of heterogeneous, dynamic and adaptive behaviour in collective systems. We use the multi-camera system described in Section 2.2 as a case study, providing insight into configuration options which arise. Specifically, we study the role of such variety in agent communication behaviour, by means of both the simulation and physical deployment of a network of smart cameras. In this section, we describe the problem and research questions considered in this paper.

3.1. Configuration Degrees of Freedom

In our case study, cameras coordinate with each other through auctions for tracked objects. Cameras participate in auctions following *auction invitations*, which are sent (or not) to other cameras selectively, based on the selling camera's *marketing strategy*. In previous work [Esterle et al. 2014] camera networks were evaluated when the cameras employed one of six possible marketing strategies for selecting which other cameras to invite to participate in an auction. Two auction initiation schedules, combined with three communication policies, give six possible marketing strategies to choose from. The auction initiation schedules are:

(1) ACTIVE, in which a camera initiates an auction for each object it owns every time it calculates the tracking performance associated with the object.
(2) PASSIVE, in which a camera initiates an auction for an object it owns when that object is about to leave its FOV.

A camera combines one of the above auction initiation schedules with one of the following communication policies:

(1) BROADCAST, which communicates the invitation to all available cameras in the network. This approach ensures all cameras which can see the object can participate (and hence buy the object), but generates a high overhead since it also includes cameras which will not respond, since they cannot see the object.

(2) STEP, which communicates the invitation to a camera if the strength of the link to that camera in the vision graph is above a certain threshold (indicating recent trading activity), otherwise inviting that camera with a very low probability.
(3) SMOOTH, which communicates the invitation to a camera with a probability based on the ratio between the strength of the link between the two cameras and that of the strongest link in its vision graph. This favours cameras with which the selling camera has traded more frequently.

However, in previous work [Esterle et al. 2014] the same marketing strategy (e.g. ACTIVE SMOOTH) was employed by all the cameras in the network for the lifetime of a deployment. In this paper, we consider the cases where (i) all cameras need not employ the same strategy as each other, (ii) each camera may vary the strategy it uses over time randomly, and (iii) each camera may learn independently which strategy to use during runtime and hence vary its strategy over time in response to environmental feedback. Following the terminology of game theory [Binmore 2007], we refer to the case when a camera uses a single strategy as a *pure strategy*. Conversely, when a camera's marketing strategy is determined at each decision point during run time according to a probability distribution, we refer to this as a *mixed strategy*. In the third case, when a camera's marketing strategy is determined at each decision point during run time by an online learning algorithm, we refer to this as an *adaptive strategy*.

However, regardless of whether the cameras use pure, mixed or adaptive strategies, at any given point in time, each camera's instantiated behaviour will be one of the six marketing strategies described above. We therefore refer to the set of marketing strategies employed by the cameras across the network at a given point in time as the configuration of the network at that time. Based on the variation in the employed marketing strategies across the network, we may describe two types of configurations:

(1) Homogeneous: A network configuration where all cameras use the same marketing strategy.
(2) Heterogeneous: A network configuration where at least two cameras use different marketing strategies.

3.2. Metrics

While cameras in the network make decisions based on local information, in common with Esterle et al. [Esterle et al. 2014], we are primarily interested in performance at the global level. This consists of two network-level measurements:

(1) *Tracking performance*, the achieved tracking performance (i.e. utility value) during a small time window for each object actually tracked (by the camera which owns it), summed over all objects.
(2) *Number of auction invitations*, the number of invitation messages sent by all cameras as a result of auction initiations, during a small time window, a proxy for communication and processing overhead.

In the simulation study a camera's utility for an object (and hence its measure of tracking performance) is simply the inverse Euclidean distance between the camera and the object. In the real camera system it is the *confidence* output of the employed SURF-based computer vision algorithm [Bay et al. 2008]. In practice the exact method used to calculate tracking performance is unimportant and we have previously explored various methods, based on a range of computer vision techniques. The number of auction invitations is simply a count of the invitation messages sent by all cameras.

While these measurements report instantaneous performance, we are interested in the online performance of the network over time. Hence each metric is the summation of the respective set of measurements over the lifetime of the deployment. We therefore

have two conflicting objectives: to maximise the tracking performance while minimising the number of auction invitations. By considering these objectives separately, we are able to obtain results in a two-dimensional objective space, which represent different points on the trade off between the two. An operator may then choose between different configurations leading to Pareto efficient outcomes, based on their relative preference between the objectives.

3.3. Research Questions

We have refined Prasath et al's [2009] key issues for engineering heterogeneity in self-organising systems, to fit the context of our smart camera network case study. Our research questions are therefore as follows:

(1) Do heterogeneous configurations enable outcomes which are more Pareto efficient than those possible in the homogeneous case?
(2) How can a decentralised network of self-interested smart cameras self-organise to a Pareto efficient configuration, given a particular scenario?

3.4. Evaluation Scenarios

For the purposes of our evaluation, a *scenario* comprises a set of cameras with associated positions and orientations, along with a set of objects and their movement paths through the environment. In this paper, we simulate and evaluate configurations within eleven qualitatively different scenarios using our open source CamSim[1] software. We also acquired video feed data from a real smart camera network, which gives us a twelfth scenario. All simulated scenarios are depicted in Figure 1, while the snapshots from the video-based scenario and associated tracking performance data are shown in Figures 2a and 2b respectively. A summary of all scenarios is given in table I. In the simulated scenarios, each object typically moves at an arbitrary but consistent speed through the environment. In those scenarios with random movement patterns, it is not possible to predict the duration each objects' visibility to each camera, since the location and angle of entry is not known and varies randomly over time. Additionally, for the scenarios with randomly generated camera layouts, cameras have different FOVs. However, for the scenarios with predefined movement paths, the length of time objects are visible is consistent and known. These are 11 time steps in scenario 9, 18 time steps in scenario 10 and between 14 and 23 time steps in scenario 11. In the real camera deployment, the objects to track were people moving at a slow walking speed.

In the simulation, the small time window used for calculating performance metrics (as described in section 3.2) corresponds to a discrete time step, and is synchronised across all cameras in the network. In our real deployment, the small time window corresponds to a single processed frame for the respective camera. In this case, the time windows of different cameras might not coincide with each other.

Unless otherwise stated, in all experiments reported in this paper, each scenario was run for 1000 discrete time steps. Due to stochasticity, 30 independent runs were conducted for each evaluation.

4. PARETO EFFICIENCY OF HETEROGENEOUS NETWORKS

Despite previous work describing six available marketing strategies [Esterle et al. 2014], they were only studied in the case when all cameras in each network used the same strategy, i.e. all the networks were homogeneous. In this section we relax this unnecessary restriction, considering the case when individual nodes (cameras) in a network can use different pure strategies from each other to govern how they adver-

[1]CamSim is available at https://github.com/EPiCS/CamSim. All scenarios are available from the repository.

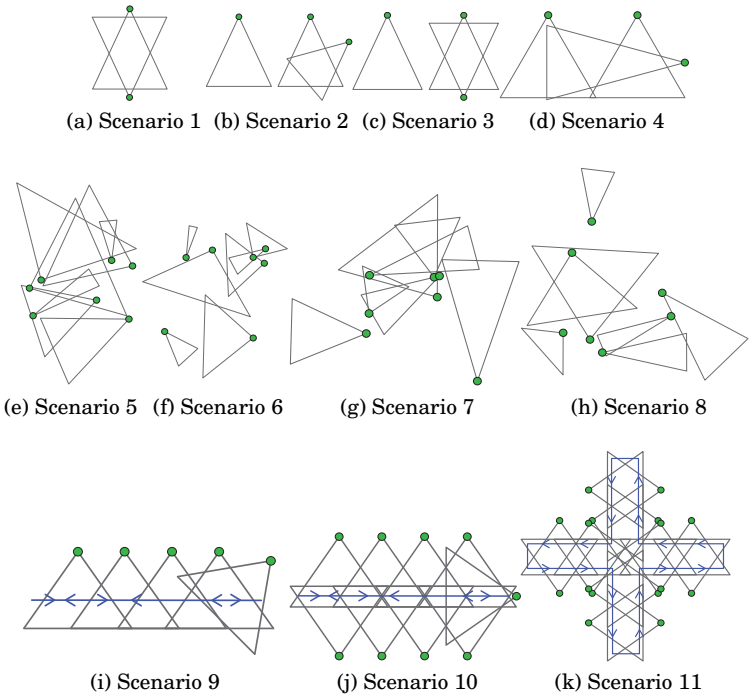


Fig. 1. The scenarios tested with our simulation tool *CamSim*. A dot represents a camera, the associated triangle represents its FOV. Blue arrows indicate the predefined movement paths.

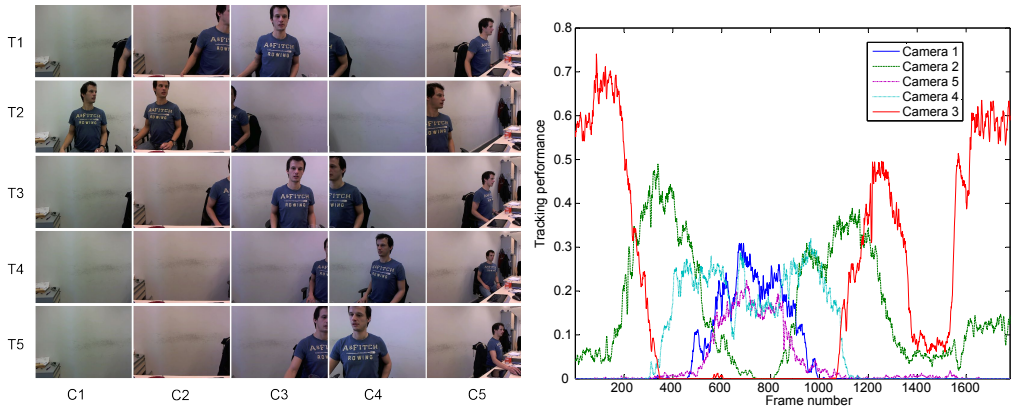


Fig. 2. Left: shots from five participating cameras tracking a single person. Right: tracking performance of each camera during the run. The performance has been smoothed using a moving average filter, where each data point is averaged over the previous five data points.

Table I. Summary of scenarios used in our study.

ID	No. of Cameras	No. of Objects	Object Movement	Timesteps	No. of Possible Configurations
1	2	4	Random	1000	36
2	3	11	Random	1000	216
3	3	4	Random	1000	216
4	3	4	Random	1000	216
5	7	9	Random	1000	$\sim 2.7 \times 10^5$
6	7	9	Random	1000	$\sim 2.7 \times 10^5$
7	7	9	Random	1000	$\sim 2.7 \times 10^5$
8	7	9	Random	1000	$\sim 2.7 \times 10^5$
9	5	3	Predefined	1000	$7,776$
10	9	1	Predefined	1000	$\sim 1.0 \times 10^7$
11	16	5	Predefined	1000	$\sim 2.8 \times 10^{12}$
12	5	1	Predefined	7120	$7,776$

Note: A random object movement path means that each object moves in a straight line until it reaches the border of the simulation area and bounces back with a randomly chosen vector. A predefined object movement path means that each object follows a predetermined path through the simulation area.

tise their auctions. Permitting this heterogeneity in the network design enables nodes to specialise to their local situation and has the effect of permitting a wider range of global outcomes than was possible in the homogeneous case. As will be shown in this section, this can lead to the global performance of the network being strictly better in terms of both the considered objectives, thus extending the Pareto efficient frontier.

However, heterogeneity itself does not necessarily lead to better outcomes. It is also possible that nodes specialise wrongly, leading to a strictly worse global outcome than was possible in any homogeneous case. Indeed, when considering all possible heterogeneous configurations for a given network, the number of configuration points increases greatly compared to the homogeneous-only case.

4.1. A Baseline Scenario

We first consider scenario 1, a baseline scenario with two cameras and four objects. Figure 3 shows the mean global performance on the two objectives, calculated over 30 independent runs. Each point represents the global outcome from one configuration κ over 1000 time steps, in terms of both metrics: its total network-wide tracking performance π and the number of auction invitations ι within the entire network. As in previous work [Esterle et al. 2014], all measured values of the different configurations are adjusted to a common scale. This normalisation of the tracking performance and auction invitations are done by the maximum achievable values, denoted π_{max} and ι_{max} respectively. Intuitively, π_{max} and ι_{max} are always obtained in a given scenario by ACTIVE BROADCAST, since this strategy always communicates to every other node at every time step, ensuring that the camera with the highest tracking performance always owns the object, but at the cost of maximal communication. This was confirmed in all our experiments. The normalised values are given by

$$\pi_{norm}(\kappa) = \frac{\pi(\kappa)}{\pi_{max}} \tag{1}$$

and

$$\iota_{norm}(\kappa) = \frac{\iota(\kappa)}{\iota_{max}}. \tag{2}$$

By enforcing homogeneity, as was done previously [Esterle et al. 2014], we have six possible deployment options. The outcomes from these homogeneous configurations

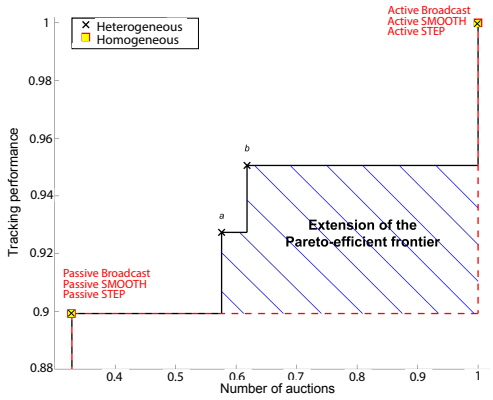


Fig. 3. Results for a baseline scenario (scenario 1) with two overlapping cameras. The original Pareto frontier when homogeneity is enforced is depicted by the dashed line. The solid line indicates the newly extended Pareto frontier when heterogeneous configurations are permitted.

are depicted as squares. In this scenario, despite the six possible homogeneous configurations, there are only two extreme observed outcomes in the objective space, one favouring each objective. This is since, in some very simple scenarios, some strategies give rise to the same communication behaviour as each other; homogeneity does not permit any more balanced outcomes in this case. However, allowing the cameras to adopt different strategies from each other introduces new possibilities. When heterogeneous configurations are included, there are 36 possible deployment options. The heterogeneous configuration outcomes are depicted as crosses.

Outcomes a and b in Figure 3 extend the Pareto efficient frontier, indicating new efficient configurations for tracking objects within the network. Additionally, both of these points lie on the newly extended Pareto frontier, since for each, no other outcome is better on both objectives. It is therefore clear from this example that heterogeneous configurations can lead to additional efficient outcomes.

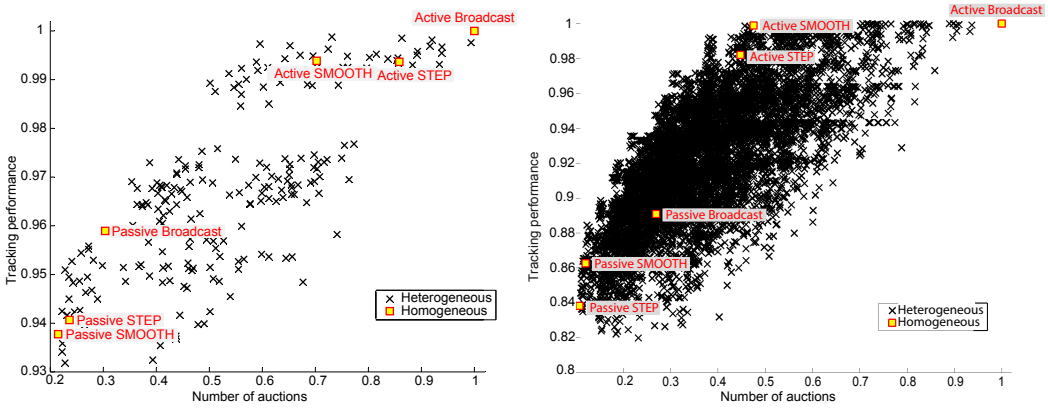
4.2. More Complex Scenarios

In this section we consider more complex scenarios. We evaluated all six homogeneous configurations in all scenarios, and all possible heterogeneous configurations in scenarios $1-9$. Due to the large number of cameras in scenarios 10 and 11, an exhaustive evaluation of all heterogeneous configurations was computationally infeasible.

Figure 4 compares outcomes from heterogeneous and homogeneous configurations in one medium sized and one large scenario, scenarios 4 and 9. In these more complex scenarios, heterogeneous configurations led to many more outcomes in the objective space. In each case, the extension of the Pareto efficient frontier brought about by heterogeneity is also apparent. However, it is also clear that the outcomes of many heterogeneous configurations are dominated, and many are strictly worse than the original outcomes from the homogeneous cases. Indeed, in all evaluated scenarios, when heterogeneous configurations of cameras are allowed, we observed system wide outcomes which both dominate and are dominated by outcomes from homogeneous configurations. In all cases, heterogeneity extended the Pareto efficient frontier.

5. PARETO EFFICIENCY OF MIXED MARKETING STRATEGIES

In Section 4 we showed the potential benefits of heterogeneity. However, we restricted the heterogeneous configurations studied to static heterogeneity: those configurations arising from cameras' varying use of pure strategies. Cameras were initialised with



(a) Scenario 4 (b) Scenario 9

Fig. 4. Performance for scenario 4 and scenario 9 showing homogeneous and heterogeneous assignment of strategies. The results have been normalized by the maximum value of the ACTIVE BROADCAST strategy and are averages over 30 runs with 1000 time steps each.

different pure strategies from each other, but did not change their strategies during run time. In this section, we consider the dynamic heterogeneous case, when cameras use mixed strategies. At each decision point in time, each camera selects one of the six pure marketing strategies according to a stationary probability distribution associated with that camera. In this case, we may consider heterogeneity at two levels. Firstly, for any distribution with a non-zero probability over more than one pure strategy, random selection will likely induce a heterogeneous configuration. Secondly, cameras may have different probability distributions from each other, governing their mixed strategies. This introduces a second level of heterogeneity.

We therefore refer to configurations where all cameras have the same distributions as *mixed*, and as *heterogeneous mixed* if at least two cameras have different distributions from each other. To select distributions, we implement a Monte Carlo approach and select random probabilities for the selection of each pure strategy at each point in time. For *mixed* strategies, each camera received the same distribution. For *heterogeneous mixed* each camera used an independent random number generator with its own (likely different) distribution.

5.1. A Baseline Scenario

We begin by studying mixed and heterogeneous mixed strategies in scenario 1. We compare our findings with the results of homogeneous and static heterogeneous configurations, which were described in Section 4.1. Figure 5 shows the outcomes from 100 uniformly sampled distributions for all cameras (*mixed strategies*) in the network and as well as 100 randomly selected distributions for each camera separately (*heterogeneous mixed strategies*). As the figure shows, enabling dynamic heterogeneity, simply by cameras altering their behaviour randomly, results in performance outcomes which extend the Pareto-efficient frontier compared to static heterogeneously and homogeneously assigned *pure* strategies. Section 8 confirms this result quantitatively.

5.2. More Complex Scenarios

This pattern is repeated across the range of scenarios tested. Figure 6 shows that in both scenarios 4 and 9 we observe that outcomes arising from the randomly sampled mixed and heterogeneous mixed strategies are typically Pareto superior to those aris-

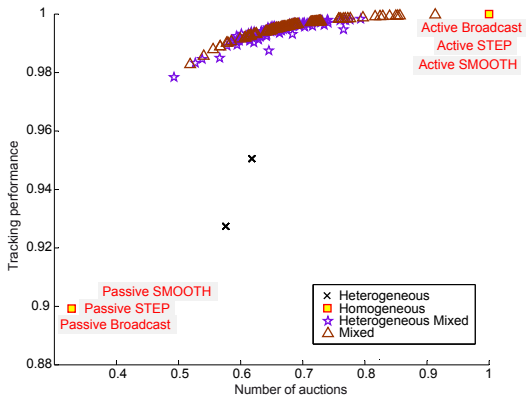
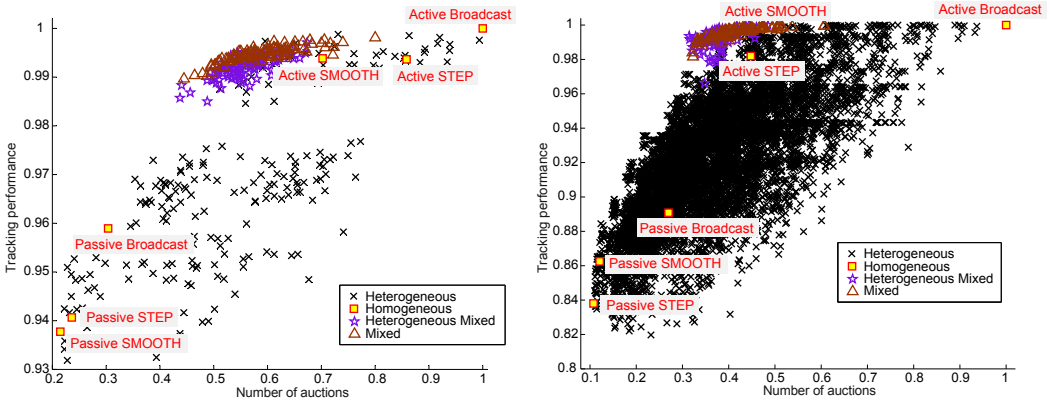


Fig. 5. Performance for scenario 1 showing homogeneous, heterogeneous, mixed and heterogeneous mixed assignment of strategies. The results have been normalized by the maximum value of the ACTIVE BROADCAST strategy and are averages over 30 runs with 1000 time steps each.

ing from both the static homogeneous and static heterogeneous cases. This perhaps surprising result suggests that dynamic configurations (i.e. those which change over time, in this case even through random behaviour) can outperform the best static heterogeneous configurations, even with the absence of any deliberate scenario-specific pattern to the dynamics. Indeed, it is further surprising that the mixed and heterogeneous mixed strategies did in no scenario generate outcomes which were at the less efficient region of the point cloud arising from the static heterogeneous configurations. Dynamics, even in the absence of any run time adaptation or advance calibration, have provided increased Pareto efficiency. As Section 8 discusses, this pattern is replicated across the majority of the evaluation scenarios studied in this paper.



(a) Scenario 4 (b) Scenario 9

Fig. 6. Performance for scenario 4 and scenario 9 showing homogeneous, heterogeneous, mixed and meta-mixed assignment of strategies. The results have been normalized by the maximum value of the ACTIVE BROADCAST strategy and are averages over 30 runs with 1000 time steps each.

5.3. Generalising Mixed Strategy Behaviour

In order to obtain some intuition behind what is being observed here, consider that for a given scenario, there must exist at least one (dynamic) configuration whose outcome is non-dominated, i.e. lies on the Pareto efficient frontier. Therefore, given sufficient advance knowledge, we could specify a non-empty set for a given scenario, in which each element is a sequence of marketing strategies for every camera at each point in time, which gives rise to a Pareto efficient global outcome. As a simple example, we might find that for the first 10 time steps of a scenario with two cameras, there are four possible Pareto-optimal optimal sets of strategy sequences for the two cameras to follow. An example of such a sequence might be as shown in Table II.

Table II. Example optimal strategy sequences for the first 10 time steps of a 2-camera scenario

Camera 1:	ASm	ASm	ASm	ASt	ASt	AB	AB	AB	ASm	PSt
Camera 2:	PB	PB	PB	ASt	ASt	AB	AB	PB	PSm	PSm

Note: AB = Active Broadcast; ASm = Active Smooth; ASt = Active Step; PB = Passive Broadcast; PSm = Passive Smooth; PSt = Passive Step

Unfortunately, determining such an optimal set of strategy sequences, even in trivial scenarios, is infeasible, and hence the above is just an example. Generally, the number of possible strategy sequence sets depends on the number of cameras, the number of strategies and the length of time to be considered. More precisely, the size of the space is s^{ct}, where s is the number of pure strategies available, c is the number of cameras and t is the number of discrete time points at which to select a configuration.

As an example, even with two cameras, six pure strategies and ten time steps, this works out as 3.6×10^{15} possible strategy sequence sets to be evaluated. And due to stochasticity in the environment (due to the uncertain presence and movement of objects) and strategies, such evaluations should be run several times to obtain statistically meaningful results. This is clearly infeasible to evaluate exhaustively, though it is likely that such a space has some structure, which may be able to be exploited through black box search techniques. However, this infeasibility implies that we are not able to compare the outcomes of our techniques against the true optimal solution in any case (e.g. as is typically done in regret calculations in reinforcement learning), as determining such an optimum is not possible in practice. We therefore rely on comparisons between the various feasible approaches which are presented.

Of particular interest here is that the optimal behaviour of the cameras will also likely be co-dependent, implying that we can analyse their behaviour game theoretically. For example, if camera 1 follows the strategy sequence in Table II then camera 2's optimal strategy is also that described in Table II. This is also the case in reverse, i.e. the cameras operating in this fashion would be in a Nash equilibrium [Binmore 2007]. However, if camera 1 deviates from the strategy sequence in Table II, then the strategy sequence for camera 2 in Table II may no longer be optimal. Indeed, an increased reward may be able to be obtained by following a different sequence. This co-dependency of strategy sequence optimality contributes an additional layer of complexity to the problem of strategy selection at the local level. More specifically, it ensures that problem faced in the online learning of such a sequence is subject to changes over time.

Given the existence of an (unknown) optimal strategy sequence, the key question in the design of an optimal heterogeneous configuration becomes: how can the system obtain global performance near to that obtained by the optimal set of strategy sequences? Given that scenarios are unknown and unpredictable, how well can different approaches for the selection of strategies at runtime approximate the optimal set of strategy sequences? In Section 4 we explored the simple approach of choosing a static

pure strategy profile for the set of cameras. In this section, we explored the approach of employing mixed strategies based on stationary probability distributions to perform this approximation. We showed that by sampling across the range of possible mixed strategy distributions, we were able to obtain highly Pareto-efficient outcomes.

However, both these approaches suffer from a critical drawback when considering their implementation in a given camera network deployment. Specifically, in order to obtain a particular preferred outcome on the Pareto-efficient frontier, e.g. one which favours reducing communication overhead, or one which balances communication and tracking performance objectives equally, we would need to know a mapping between (mixed) strategy distributions across all cameras and their corresponding outcomes in the objective space. Such a mapping would need to be determined offline in advance, and would require *a priori* knowledge of the scenario, including object movements, camera positions and dynamics of the network itself. The lack of precisely this information is a key assumption and motivator of the online socio-economic approach studied in this paper, and hence violating this assumption in order to select an appropriate mixed strategy profile would not be appropriate. In other words, if such information was indeed available in advance, then other offline calibration methods may be pursued instead of the one studied in this paper. For this reason, we do not pursue this offline analysis further. In order for a strategy profile selection technique to be of practical use within this online scenario, where scenario information is not known in advance, such a mapping from (mixed) strategy distributions to outcome positions would have to be highly robust to differences between scenarios. There is no evidence that this is the case; indeed, due to the complex interactions between the effects of different strategies observed here and in prior work [Esterle et al. 2014; Esterle et al. 2012], this is unlikely to be true. This means that a new mapping would have to be learnt for each specific deployment, which is tantamount to performing offline calibration. It is also unclear how this might generalise to the case when new or different base strategies are available, as might generally be the case. Therefore, in the following section we instead develop and explore online learning as an alternative approach, where the assumptions of no advanced scenario knowledge can be maintained.

6. DECENTRALISED ONLINE LEARNING OF PARETO EFFICIENT CONFIGURATIONS

Section 4 showed that by permitting heterogeneous configurations of nodes, global outcomes may be obtained which are more Pareto efficient than in the homogeneous case. Section 5, further showed that in addition to heterogeneous configuration, dynamic configurations, those where cameras change their strategies during runtime, can lead to a further increase in Pareto efficiency. In Section 5.3, we discussed the difficulties in choosing a particular static or dynamic heterogeneous configuration at any point in time, in order to achieve a desired efficient global outcome, in a particular scenario.

This could be tackled as an offline search problem, as part of multi-camera calibration. However, doing so would assume that we know the characteristics of the scenario in advance, including camera placement and orientation, expected object movement patterns and any runtime failures or additions (such as studied in [Esterle et al. 2012]). Therefore, we instead tackle this problem by extending the idea followed in our previous work, where individual cameras learn behaviours online during run time.

In this manner, a camera's strategy selection is made autonomously using a learning technique at the local level, which provides adaptation at run time based on feedback from locally observed metrics: in this case the number of auction invitations sent by the node and its tracking performance (as opposed to the equivalent metrics for the network as a whole). We are then interested in observing the effect of this parallel local learning within cameras on the metrics at the network level.

This purely online approach ensures that the deployment of a multi-camera system remains simple and does not require advance calibration. By using online learning, in this section we show that camera networks are able to achieve Pareto-efficient outcomes, tunable according to the operator's preference, without the need to consider the nature of the deployment scenario in advance.

6.1. Learning Efficient Configurations using Bandit Solvers

From the perspective of an individual camera, its task is to select a marketing strategy from those available, at each point in time, such that it maximises its expected tracking performance while minimising its auction overhead, over time. As described in Section 5.3, there will exist an ideal sequence of marketing strategies for each camera, such that the network as a whole achieves Pareto-efficiency, however each camera cannot know this sequence in advance. Furthermore, as cameras learn, the learning problem facing the other cameras changes in response. We therefore consider that a camera is faced with a non-stationary online algorithm selection problem [Gagliolo and Schmidhuber 2011]. Our approach is to consider this as an variant of the multi-armed bandit problem [Auer et al. 2002]. This problem is analogous to being faced with n slot machine arms, where each pull of an arm returns a random reward drawn from an unknown distribution associated with that arm. Given m total arm pulls, the task is to select which arms to pull such that the total reward obtained is maximised. If the player knew the distributions behind each arm, then he could simply select the best arm for every pull. However, since the distributions are unknown, he must sample from each arm in order to learn its reward distribution. The multi-armed bandit problem therefore encapsulates the classic *exploration vs exploitation* dilemma. However, some of the assumptions present in the classic multi-armed bandit problem formulation may not be appropriate in this setting. Firstly, the reward distributions are usually assumed to be stationary over time, and secondly it is assumed that the bounds on the obtainable rewards are also known. Neither assumption can be made in our case.

Nevertheless, the bandit framework is a useful model, where each marketing strategy can be considered an arm of a bandit. Each camera node can choose to use one strategy (i.e. pull an arm) at each time step, and can receive a resulting reward, derived from its local metrics. In this way, a camera learns which strategy performs well in its current situation within the scenario, and exploit that knowledge to maximise its performance. There are a number of so-called bandit solving algorithms to be found in the literature. In this paper we consider three well known bandit solvers: the simple EPSILON-GREEDY [Sutton and Barto 1998], UCB1, which is known to perform well in stationary problems [Auer et al. 2002], and SOFTMAX [Sutton and Barto 1998]. Of these, EPSILON-GREEDY requires an ϵ value to determine the amount of exploration, UCB1 requires no parameters, while SOFTMAX uses a temperature value to govern how an arm's expected reward influences its probability of selection.

Epsilon greedy [Sutton and Barto 1998] is perhaps the simplest bandit solver. First we try each action (or arm) j, recording the reward obtained. Thereafter, the arm with the greatest average reward $\bar{x}_j$ is selected with probability $1 - \epsilon$; with probability ϵ a random arm is chosen. Each time an arm is chosen, the average reward associated with that arm, $\bar{x}_j$ is updated according to the obtained reward.

The SOFTMAX action selection method [Sutton and Barto 1998] aims to improve upon EPSILON GREEDY by varying arm selection probabilities according to the estimated value of each arm. The technique selects an arm according to a probability $p(j)$ associated with each arm j, determined by Equation 3. The method uses a "temperature" parameter T, which determines the uniformity with which arms with different expected rewards $\bar{x}_j$ get selected, lower T making selection focus on arms with high $\bar{x}_j$.

$$p(j) = \frac{e^{\bar{x}_j/T}}{\sum_{k=1}^{n} e^{\bar{x}_k/T}} \tag{3}$$

For the UCB1 strategy [Auer et al. 2002], which uses confidence bounds, we first try each arm j, and record the reward $\bar{x}_j$ arising from that arm, setting n_j to 1. Thereafter, we select an arm with the index as given by Equation 4, updating the average reward $\bar{x}_j$ observed from each arm, as well as the number of times n_j an arm j has been tried so far. The total number of arms tried so far is given by n.

$$\underset{j}{\operatorname{argmax}}\, \bar{x}_j + \sqrt{\frac{2 \ln n}{n_j}} \tag{4}$$

In applying bandit solvers to algorithm selection at the local level in a self-organising system, we must define local reward functions, such that the global system's objectives are achieved. In this case, this is further complicated by the presence of multiple objectives at the global level, and corresponding multiple metrics locally at each node. A common approach to the definition of a reward function in multi-objective learning is scalarisation of multiple reward signals into a single reward value, although the choice of scalarisation technique can have a significant impact on the ability of the learning algorithm to properly explore the Pareto frontier [Vamplew et al. 2008]. Various scalarisation methods have been proposed [Van Moffaert et al. 2013b; 2013a], however at this stage we are more concerned with the overall effect of decentralised online learning of marketing strategies and leave the question of which scalarisation technique should be preferred as an item for future research. Additionally, preliminary experiments showed that the choice of bandit solver algorithm had far greater effect on the outcome than the choice of scalarisation technique.

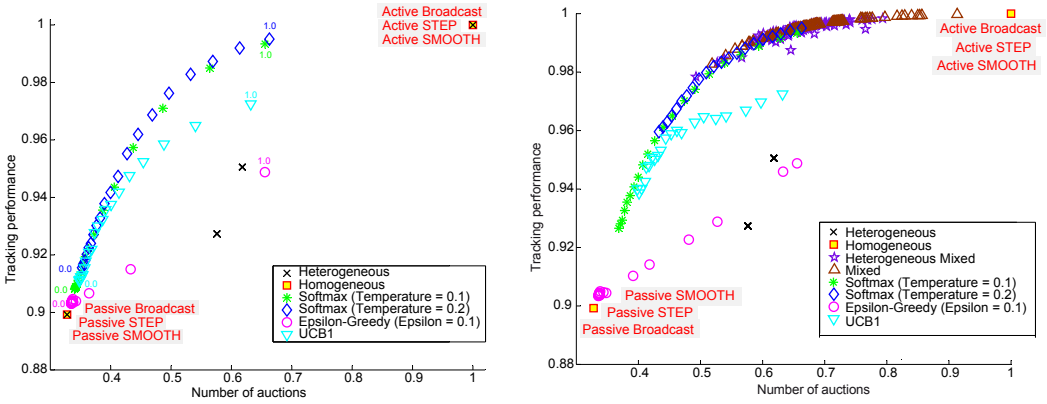
Therefore, in this paper we use the usual linear combination approach to scalarisation of the local metrics:

$$\textit{reward} = \alpha \times \textit{total camera utility} - (1 - \alpha) \times \textit{auction invitations} \tag{5}$$

where *total camera utility* is the camera's total utility value as found in previous work [Esterle et al. 2014], which sums the tracking performance of (i.e. the utility obtained from) all objects tracked by this camera, plus the camera's balance of payments from all trading activity during this time step. The number of auction invitations sent by this camera at this time step is denoted by *auction invitations*. α allows us to change the camera's preference in favour of either maximising tracking performance or minimising the number of auction invitations. Therefore, α may be used as a handle with which to guide local learning such that outcomes at the global level favour appropriate regions of the Pareto efficient frontier.

Figure 7a shows the outcomes in scenario 1, when configurations learnt using bandit solvers are compared with static homogeneous and heterogeneous configurations. For EPSILON-GREEDY, ϵ values of 0.1, 0.01 and 0.001 were tried. In all scenarios, with 1000 time steps, $\epsilon = 0.1$ obtained the most Pareto efficient outcomes and is therefore used in all results in this paper. Outcomes are shown for EPSILON-GREEDY, UCB1, and SOFTMAX, the latter with temperature values 0.1 and 0.2. For each bandit solver, results are shown when α is varied between 0 and 1 in intervals of size 0.05.

The results in Figure 7a clearly show that configurations found with bandit solvers provide many more outcome points in the objective space than were possible in the static homogeneous and heterogeneous cases, and that many of these outcomes are highly Pareto efficient. Even though we presented the static heterogeneous configura-



(a) Using original reward values for bandit solvers. (b) Using reward values normalised by distribution.

Fig. 7. Performance for scenario 1 showing homogeneous and heterogeneous assignment of strategies as well as assignments done by bandit solvers. In (b), results from mixed and heterogeneously mixed strategies are also shown. The results have been normalized by the maximum value of the ACTIVE BROADCAST strategy and are averages over 30 runs with 1000 time steps each. Additionally, in (b) the bandit solvers' reward functions normalised the number of auction invitations by distribution at runtime (see Section 6.2).

tion outcomes exhaustively, bandit solvers were able to obtain system wide outcomes which extend the Pareto efficient frontier obtained in the static heterogeneous case.

As with mixed strategy outcomes as presented in Section 5, dynamic configurations arising from online learning, even with no advance scenario knowledge, can outperform even the best static heterogeneous configurations.

6.2. Camera Level Normalisation by Distribution

Section 6.1, showed how bandit solvers can be used within cameras in order to select marketing strategies during runtime, appropriate to the scenario and the node's preferences between objectives (in this case its α value). We also showed that by varying the α value, global outcomes ranged over the Pareto efficient frontier. However, it is also clear from Figure 7a that the results from the bandit solvers cluster towards the lower left of the Pareto front, while outcomes in the upper right are more thinly spread.

This bias in outcomes is a result of the nature of the observed metrics at the camera level, and their combination in the local reward function. Ideally, α would be used to weigh the two objectives evenly, such that the outcome position on the Pareto frontier can be determined directly by setting α. E.g. an α value of 0.25 would lead to an outcome value 25% of the way along the length of the achieved front. In order to achieve this, we would need to normalise the two components of the reward function. However, although a camera knows its own tracking performance associated with an object, it cannot know what payment it might have received, had it advertised the object to a camera which it did not. The upper bound on the camera's utility is therefore not known, and will vary significantly with every time step. Nevertheless, we are still able to mitigate the bias effect somewhat by attending to the second component of the reward function, the number of auction invitations issued by the camera. The upper bound on this value will also vary, but in this case only with the number of objects and other cameras currently known to the camera.

We are therefore able to perform some estimated normalisation of the number of auction invitations at the local level. Figure 8 shows the frequency with which cameras send auction invitations to other cameras over time. Clearly, cameras are less com-

municative more often than they are more communicative. As it turns out, this skew in the distribution appears to have a large effect on the bias observed in the outcome Pareto front. We are able to account for this skew effect by introducing a normalisation by distribution process into the auction invitation component of the local reward.

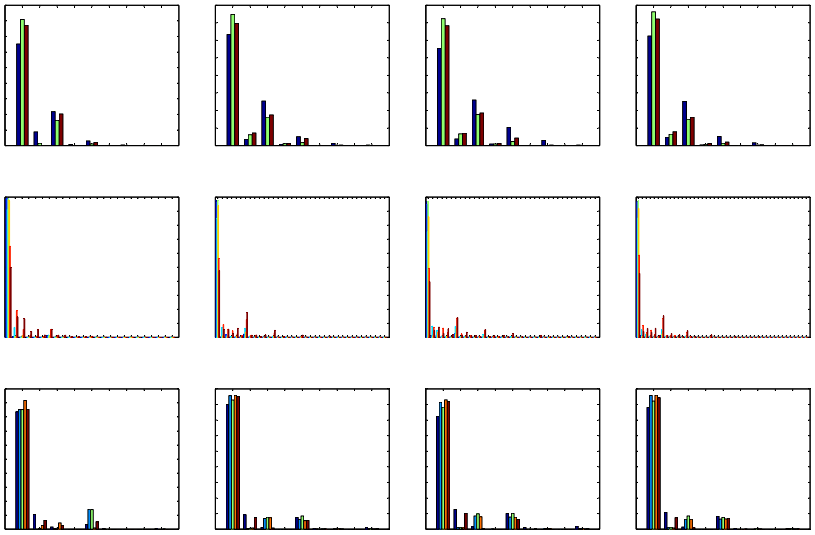


Fig. 8. A selection of histograms showing the frequency distribution of auction invitations sent per time step. The bins on the x axis represent 0, 1, 2, 3 and 4 auctions per time step from left to right. The y axis shows the frequency with which this number of auctions was observed in each camera. Each camera is represented by one bar per bin. Each camera is represented using one bar per bin. From top to bottom, the rows show scenarios 3, 6 and 9 respectively. From left to right, the columns show results when cameras use EPSILON-GREEDY, SOFTMAX with temperature = 0.1, SOFTMAX with temperature = 0.2 and UCB1 respectively. In all cases $\alpha = 0.5$.

More specifically, each camera records the value of *auction invitations* for each time step throughout its lifetime. When a new value is observed, its rank within the historical values is calculated, and then scaled to be between 0 and 1. The original value is then added to the historical record. For example, if the new value is greater than the largest observed value so far, its normalised value is 1. Similarly, if a new value falls half way along the list of historically observed values, its normalised value is 0.5.

By normalising in this way, the skew present in the distribution of original values is reduced, and we obtain a more even spread of outcomes along the achieved frontier. Figure 7b shows this for scenario 1, and can be compared with Figure 7a. The results from mixed and heterogeneous mixed strategies, described in Section 5 have also been added back in for comparison. A bias, though less pronounced, is still present with EPSILON-GREEDY and UCB1. This skewed distribution pervades all scenarios we evaluated, therefore we adopted this normalisation method in all subsequent experiments.

For a quantitative evaluation of the effect of the skew observed here, we measured the distance between each pair of consecutive points on the Pareto frontier, for each bandit solver. From this, we extracted the mean distance between pairs of points, as well as the variance about the mean. These figures tell us both how far apart and how evenly spaced the achieved points on the front are. Table III shows these values for the non-normalised and normalised by distribution versions of the bandit solvers. In each

Table III. A comparison of distances between pairs of adjacent outcome points on the Pareto frontier, for both normalised and non-normalised cases, for each of the bandit solvers. Mean pairwise distances and variances about these means are shown, averaged over 30 runs. Statistically significant differences are shown in bold, according to rank sum tests calculated at the 95% confidence level.

	EPSILON GREEDY		SOFTMAX $T = 0.1$		SOFTMAX $T = 0.2$		UCB1	
	Original	Normalised	Original	Normalised	Original	Normalised	Original	Normalised
Mean	0.1944	0.1533	0.0879	**0.0495**	0.0611	**0.0287**	0.0821	0.0795
Variance	0.0768	0.0577	0.0066	**0.0010**	0.0017	**0.0003**	0.0040	**0.0020**

case, the values reported are again averaged, as we take the mean over 30 independent runs. As can be seen from the table, the average distance between achieved points is consistently lower in the normalised case. More importantly from our point of view however, is that the variance is also substantially lower in all cases. This tells us that the achieved points are more evenly spread in the normalised case.

6.3. Learning in More Complex Scenarios

Figure 9 shows results for scenarios 4, 5, 8 and 9. In all cases, bandit solvers used the normalisation by distribution method described in Section 6.2. In each of these more complex scenarios, bandit solvers appear to be able to obtain outcomes which extended the Pareto efficient frontier of the exhaustively evaluated static configurations. This is particularly true of SOFTMAX and UCB1, all of which obtained a range of highly Pareto efficient outcomes. On further evaluation, as will be reported in Section 8, a statistically significant Pareto frontier extension is observed to arise from online learning in all scenarios except 2 and 12. In those cases, the outcomes from learning compare well with the most efficient static heterogeneous configurations.

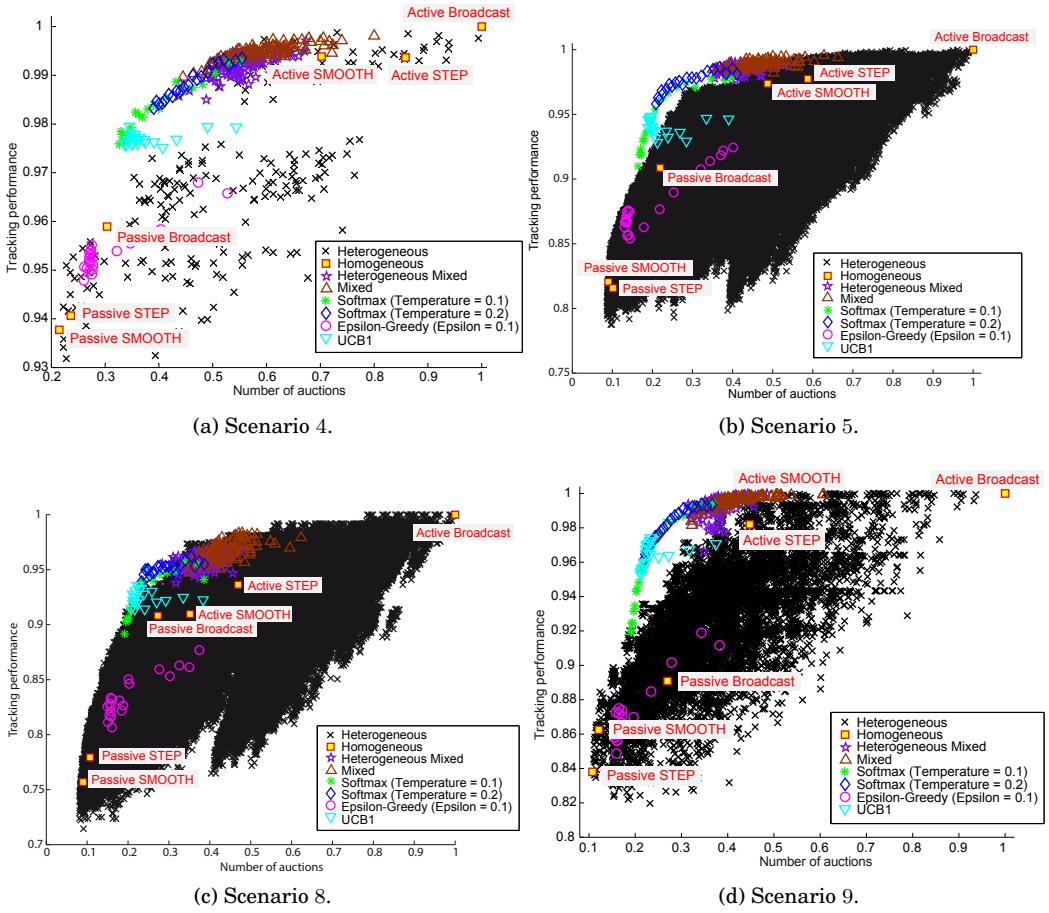
The spread of outcomes can be observed to vary depending on the particular scenario and the choice of bandit solver employed. The bias associated with EPSILON-GREEDY, and to a lesser extent UCB1 remains. Outcomes from SOFTMAX are more evenly spread as α varies, typically obtaining the most even spread across the frontier in each scenario, especially when compared with outcomes arising from the mixed and heterogeneous mixed strategies. Section 8 also contains a quantitative evaluation of the evenness of the spread associated with the outcome points.

A further observation is that the outcomes from the bandit solvers never reach either extreme of the objective space, instead gravitating toward the middle. This is due to exploration behaviour: since we are measuring online performance, in order to achieve extreme outcomes, the network would need to use the required configuration throughout its lifetime.

7. REAL CAMERA NETWORK RESULTS

We furthered our evaluation using video feed data from a real smart camera network; this is referred to as scenario 12 in Table I. A SURF-based tracking approach [Bay et al. 2008] was used to detect and track a person within the network of cameras. The SURF-based approach initially extracts features from a model image and tries to re-identify these model features in the consecutive frames. The percentage of re-identified features represents the confidence (cf. Section 2.2) of our SURF-based tracker. Figure 2a shows snapshots from each camera at five different points in time (left) and the tracking performance of each camera over time (right). Each camera captured 1780 frames, looped four times to create a total of 7120 frames, each with a resolution of 640×480. When PASSIVE strategies were employed, auctions were initiated when the tracked object was within 20 pixels of the border of the FOV.

Figure 10 shows the results obtained from all static and dynamic homogeneous and heterogeneous strategies as well as those obtained by decentralised online learning. As with the results in Section 4, heterogeneous configurations lead to system wide



(a) Scenario 4. (b) Scenario 5.

(c) Scenario 8. (d) Scenario 9.

Fig. 9. Performance for scenarios 4, 5, 8 and 9, showing homogeneous and heterogeneous assignment of strategies, mixed and heterogeneous mixed strategies, and assignments done by bandit solvers. The results have been normalized by the maximum value of the ACTIVE BROADCAST strategy and are averages over 30 runs with 1000 time steps each. The bandit solvers' reward functions normalised the number of auction invitations by distribution at runtime.

outcomes which are more Pareto efficient then those possible in the homogeneous case. Furthermore, as with the results in Section 6, the use of decentralised online learning of marketing strategies also extended the Pareto efficient frontier when compared to the homogeneous case. As with some of the simulation scenarios, in this instance learning was not able to generate outcomes dominating the most Pareto efficient heterogeneous cases, however they do compare well here also.

8. QUANTIFYING PARETO EFFICIENCY

We quantitatively compared the quality of Pareto frontiers using three indicators, within a relevant region of the objective space. More specifically, the span[2] of objective values for both objectives is determined for each run and used to scale all outcomes pertaining to that run. For each simulation run, the set of outcomes comprising all the

[2]Given all outcomes from the studied approaches for any run, we refer to the difference between the maximum and minimum values in these outcomes for an objective the *span* for that objective for the run.

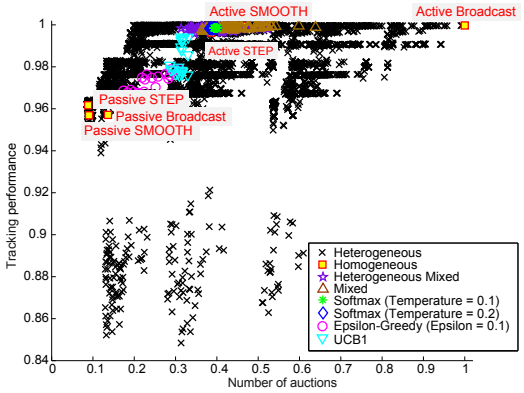


Fig. 10. Performance of all configurations on scenario 12, showing homogeneous and heterogeneous assignment of strategies, mixed and heterogeneous mixed strategies, and assignments done by bandit solvers. The results have been normalized by the maximum value of the ACTIVE BROADCAST strategy and are averages over 30 runs with 1000 time steps each. The bandit solvers' reward functions normalised the number of auction invitations by distribution at runtime.

static configurations contain the extreme values for both our objectives, and thus provides the span to be used as the scaling factor[3]. Pareto frontier indicators are computed using the frontiers resulting from all studied approaches in this scaled objective space. This enables us to compare Pareto frontiers in relation to each other fairly, as opposed to each being computed and compared with respect to an arbitrary scale. Moreover, we perform rank sum tests on the quality of frontiers.

8.1. Notation

The Pareto efficient frontier considering the outcomes from static homogeneous configurations gives us a frontier which we call h. The Pareto efficient frontier considering all the outcomes from both static homogeneous and static heterogeneous configurations gives us a frontier that we call h-he. Outcomes due to sampling of mixed strategies, as described in Section 5, together with homogeneous configuration outcomes we call h-m and h-hm for mixed and heterogeneous mixed strategies respectively. Accordingly, h-he-m and h-he-hm refer to the fronts arising from the combinations of homogeneous, static heterogeneous and either form of mixed strategies respectively. Outcomes due to a bandit solver, together with static homogeneous configuration outcomes, give us frontiers h-eg, h-sm, or h-ucb, depending on the considered bandit solver being EPSILON-GREEDY, SOFTMAX, or UCB1 respectively. We focussed this analysis on SOFTMAX with a temperature value of 0.2, since this obtained the lowest variance in Table III, indicating more evenly spread outcome points. Outcomes due to a bandit solver, together with static homogeneous configuration outcomes and static heterogeneous configuration outcomes, give us frontiers h-he-eg, h-he-sm, or h-he-ucb. Similarly, fronts arising from the combination of both mixed and learnt strategies are termed e.g. h-he-m-eg.

8.2. Hypervolume

One way of comparing Pareto efficient frontiers is to compute the hypervolume [While et al. 2006] under each frontier, given a reference point. This is particularly appropri-

[3]Scenarios 10 and 11 admit a combinatorially large number of heterogeneous configuration, which were not possible to simulate. Thus, the outcomes from heterogeneous configuration remain unknown to us for these scenarios. The set of static homogeneous outcomes still contain the extreme values for both our objectives per run, therefore we used this set to determine the span across both objectives in these scenarios.

ate when the true frontier is not known, as in this case, and as such the regret measure cannot be used [Vamplew et al. 2011]. The reference point can be specified as the vector of worst case values in the scaled objective space. Thus, a tracking performance value of 0.0, and a number of auction invitations value of 1.0 in the scaled space specifies our reference point. The greater the hypervolume of a Pareto frontier, the more efficient it is. In this case, we calculate the online hypervolume [Vamplew et al. 2011], since we are interested in online performance, that calculated as a sum over time.

Table IV shows the medians (across 30 independent runs) of the hypervolumes of the aforementioned frontiers, indicating statistical significance in the extensions with respect to the fronts arising from non-learnt configurations, across the scenarios considered in this paper. Firstly, it is evident that heterogeneity of marketing strategies, brought about through either static configurations (h-he) or mixed strategies (h-m and h-hm), extend the homogeneous frontiers h, regardless of the scenario. Statistically significant differences were observed in all scenarios except scenario 10. Secondly, in roughly half of the scenarios studied, mixed strategies extended the Pareto frontier relative to static heterogeneous configurations. This is shown by the presence of the letter b in the columns h-he-m and h-he-hm. Turning to the outcomes from the learnt strategies, we thirdly note that the frontiers h-eg, h-sm, and h-ucb often extend the Pareto frontier arising from homogeneous configuration outcomes h. This is shown by the frequent presence of the letter a in those columns. Though statistically significant differences between these two cases were not observed in all cases, the median hypervolume values were always observed to be larger in the case which included learning. Moreover, the frontiers arising from homogeneous, mixed and learnt strategies often further extend the frontiers h-m and h-hm, showing that learning often provided increased Pareto efficiency over the two forms of mixed strategies studied.

Importantly, when compared against both static homogeneous and static heterogeneous configuration outcomes, the addition of learning further extends the frontier in many cases. This is particularly true of SOFTMAX and UCB1. EPSILON GREEDY did not perform as well. Also, scenarios 2 and 12 proved more challenging for the bandit solvers. From our indicative graphical results (e.g. Figure 9b), we speculate that this ability of learning to extend the Pareto frontier relative to the full set of static heterogeneous configurations may more widely be found. However, the computational infeasibility of exhaustively evaluating all static heterogeneous configurations on the larger scenarios prevented us from obtaining sufficient results for statistical testing. Despite this, it is clearly shown that decentralised online learning, based on bandit solvers, can lead to the network self-organising towards global outcomes that are more Pareto efficient than those from static heterogeneous configurations in many scenarios. Figure 9d provides an intuitive visualisation of this. The adaptive nature of the dynamics generated by the bandit solvers, based on feedback from the environment, also leads to higher Pareto efficiency than the non-adaptive dynamic case, as exemplified by the mixed strategies. Although mixed strategies typically achieve a highly efficient set of outcomes, they are often highly clustered. Without a parametrisable feedback signal, as provided by the bandit solvers, it is not possible to achieve a directed search of the Pareto frontier during run time. This leads to lost efficiency in the non-adaptive case.

In summary, Table IV shows i) that heterogeneity clearly increases Pareto efficiency compared to homogeneity, and ii) that in many cases adaptive configurations induced by online learning allow the network to reach favourable parts of the outcome space, which are inaccessible in any static case. When learning does not lead to a frontier which represents an extension to the Pareto frontier, it instead typically finds outcomes which are comparable with the most efficient static heterogeneous configurations. Considering that at deployment time one does not have the luxury of identifying which static heterogeneous configurations to choose from an exponentially growing

Table IV. Medians over 30 independent runs of the hypervolume of the Pareto front resulting from various configurations: homogeneous (h), homogeneous + heterogeneous (h-he), homogeneous + mixed (h-m, h-hm), homogeneous + heterogeneous + mixed (h-he-m, h-he-hm), homogeneous + learnt (h-eg, h-sm, h-ucb), homogeneous + heterogeneous + learnt (h-he-eg, h-he-sm, h-he-ucb), homogeneous + mixed + learnt (h-m-eg, h-m-sm, h-m-ucb, h-hm-eg, h-hm-sm, h-hm-ucb), homogeneous + heterogeneous + mixed + learnt (h-he-m-eg, h-he-m-sm, h-he-m-ucb, h-he-hm-eg, h-he-hm-sm, h-he-hm-ucb)

Scenario	Pareto fronts without learnt outcomes											
	h		h-he		h-m		h-hm		h-he-m		h-he-hm	
1	0.0		0.30929	a	0.68673	ab	0.72107	abce	0.68673	ab	0.72107	abce
2	0.80137		0.90698	acd	0.8554	a	0.85584	a	0.91055	acd	0.9109	acd
3	0.72707		0.8767	acd	0.83887	a	0.84352	a	0.88769	acd	0.89104	abcd
4	0.65202		0.84816	acd	0.81301	a	0.81993	a	0.8616	abcd	0.86327	abcd
5	0.82796		–		0.88715	a	0.88698	a	–		–	
6	0.9383		–		0.95145	a	0.95204	a	–		–	
7	0.80726		–		0.86218	a	0.86313	a	–		–	
8	0.78658		–		0.86931	a	0.88135	ac	–		–	
9	0.85205		0.94012	acd	0.90023	a	0.90077	a	0.94583	abcd	0.94699	abcd
10	0.89392		–		0.905		0.91355		–		–	
11	0.68725		–		0.8487	a	0.83122	a	–		–	
12	0.93062		0.98824	acd	0.93615	a	0.94699	ac	0.98824	acd	0.98824	acd
	Pareto fronts with learnt outcomes											
	h-eg		h-sm		h-ucb		h-he-eg		h-he-sm		h-he-ucb	
1	0.21849	a	0.75989	abcdef	0.65098	ab	0.35997	ab	0.75989	abcdef	0.65098	ab
2	0.84635	a	0.83036	a	0.81477		0.90866	acd	0.90874	acd	0.90705	acd
3	0.79252	a	0.83807	a	0.83109	a	0.88277	acd	0.8888	abcd	0.89181	abcd
4	0.72772	a	0.82902	a	0.80629	a	0.85151	acd	0.8694	abcd	0.86682	abcd
5	0.85765	a	0.90883	acd	0.88058	a	–		–		–	
6	0.95079	a	0.9495	a	0.95023	a	–		–		–	
7	0.83062	a	0.83815	a	0.83134	a	–		–		–	
8	0.82619	a	0.87712	a	0.86225	a	–		–		–	
9	0.88409	a	0.93545	acd	0.92265	acd	0.94012	acd	0.95475	abcdef	0.94842	abcd
10	0.91748		0.91667		0.92355	a	–		–		–	
11	0.72752		0.82322	a	0.80153	a	–		–		–	
12	0.95727	acd	0.93141		0.95302	acd	0.98835	acd	0.98824	acd	0.98824	acd
	h-m-eg		h-m-sm		h-m-ucb		h-hm-eg		h-hm-sm		h-hm-ucb	
1	0.71112	abce	0.78301	abcdef	0.79487	abcdef	0.74209	abcdef	0.78412	abcdef	0.80311	abcdef
2	0.88807	acd	0.85841	a	0.85643	a	0.88674	acd	0.85708	a	0.8578	a
3	0.87684	acd	0.84971	a	0.85532	acd	0.88291	acd	0.85055	a	0.85874	acd
4	0.84425	ac	0.84063	acd	0.86016	acd	0.84687	acd	0.84682	acd	0.86449	abcd
5	0.9063	acd	0.91837	acd	0.91628	acd	0.90669	acd	0.9175	acd	0.91576	acd
6	0.96393	acd	0.95445	a	0.95774	acd	0.96573	acd	0.95533	a	0.95892	acd
7	0.88647	acd	0.86544	a	0.87142	a	0.88306	acd	0.86448	a	0.87456	a
8	0.89729	acd	0.89777	acd	0.90003	acd	0.90689	acd	0.89841	acd	0.90452	acd
9	0.91474	acd	0.93618	acd	0.93399	acd	0.91659	acd	0.93651	acd	0.93553	acd
10	0.92407	a	0.91691		0.92446	a	0.92851	a	0.91722		0.9275	a
11	0.8675	acd	0.85478	ad	0.86499	acd	0.85108	a d	0.83186	a	0.84451	a d
12	0.96034	acd	0.93617	a	0.9541	acd	0.96355	acd	0.94699	ac	0.95551	acd
	h-he-m-eg		h-he-m-sm		h-he-m-ucb		h-he-hm-eg		h-he-hm-sm		h-he-hm-ucb	
1	0.71112	abce	0.78301	abcdef	0.79487	abcdef	0.74209	abcdef	0.78412	abcdef	0.80311	abcdef
2	0.91135	acd	0.91059	acd	0.91071	acd	0.91183	acd	0.91166	acd	0.91106	acd
3	0.89424	abcd	0.89196	abcd	0.89805	abcd	0.89685	abcd	0.89293	abcd	0.89937	abcd
4	0.86704	abcd	0.87183	abcd	0.87753	abcdef	0.86683	abcd	0.87153	abcd	0.87655	abcdef
5	–		–		–		–		–		–	
6	–		–		–		–		–		–	
7	–		–		–		–		–		–	
8	–		–		–		–		–		–	
9	0.94583	abcd	0.95542	abcdef	0.95337	abcdef	0.94699	abcd	0.95538	abcdef	0.95453	abcdef
10	–		–		–		–		–		–	
11	–		–		–		–		–		–	
12	0.98835	acd	0.98824	acd	0.98824	acd	0.98835	acd	0.98824	acd	0.98824	acd

Note: The Wilcoxon rank sum test was used with a 95% confidence level to assess statistical significance. The following symbols denote significant increase in hypervolume w.r.t. a front without learnt outcomes: "a" w.r.t. front h, "b" w.r.t. front h-he, "c" w.r.t. front h-m, "d" w.r.t. front h-hm, "e" w.r.t. front h-he-m, "f" w.r.t. front h-he-hm. Tests that were not performed due to computational infeasibility are denoted by "–".

set of alternatives, online learning, which requires no *a priori* scenario knowledge, presents a substantially more attractive technique.

8.3. Spread

Hypervolume tells us the Pareto efficiency of the frontiers arising from each of the approaches evaluated. However, operators may have varying preferences between the two objectives considered here. It is therefore important that the approach is also able to provide a good spread of possible outcomes across the objective space. In order to measure the spread of the outcomes in the objective space, we firstly used the *inverted generational distance (IGD)* [Coello and Cruz Cortés 2005; Zhang et al. 2008] indicator. This indicator measures the average minimum distance of points in the *true Pareto frontier* from the obtained frontier:

$$IGD(P_o) = \frac{\sum_{i=1}^{|P_t|} d_i(P_o)}{|P_t|}, \tag{6}$$

where, P_o is the obtained frontier, P_t is the true frontier, and $d_i(P_o)$ is the minimum Euclidean distance of point i from the obtained frontier P_o.

If the obtained frontier is such that points in this frontier lie close to the points in the true frontier, the IGD is small. We do not know the true Pareto frontier in our case, but we can approximate this by specifying a *reference set* that is widely and evenly spread across the part of the objective space that can never be dominated by an obtained frontier. However, frontiers can be evenly spread across the objective space yet be further away from many points in the reference set, and vice versa. Both these situations result in low IGD values, limiting its usefulness. The IGD indicator alone is therefore not enough to judge the evenness in the spread of outcomes with respect to the reference set. We thus secondly quantify the spread by computing the *variance* in minimum distances $d_i(P_o)$ (see Equation 6), between points in the reference set and the obtained frontier. We call this indicator $IGDV$:

$$IGDV(P_o) = \frac{\sum_{i=1}^{|P_t|}(d_i(P_o) - IGD(P_o))^2}{|P_t|}. \tag{7}$$

A smaller $IGDV$ indicates the minimum distances between all the points in the reference set and obtained frontier to be generally similar. Therefore, if the reference set is evenly spread across the objective space, a small $IGDV$ value indicates evenness in spread. In essence, $IGDV$ is the more reliable indicator for determining the evenness in spread of a frontier.

Table V shows the comparisons of the IGD and $IGDV$ indicators between Pareto frontiers with only homogeneously mixed (m), only heterogeneously mixed (hm), only EPSILON GREEDY (eg), only SOFTMAX (sm), and only UCB (ucb) outcomes, respectively. For the reference set P_t, we choose 501 evenly spread points $\in \mathbb{R}^2$ along the axes, 250 points $\in [(0,0),(0,1))$, 250 points $\in ((0,1),(1,1)]$, and the dominant point $(0,1)$. We choose this reference set because we know the extreme points of our obtained frontiers, which reside in the static frontiers, and we normalise all outcomes (per run) using such extreme points (per run), to lie in the first quadrant (unit square).

In terms of IGD, eg consistently outperformed both m and hm across the scenarios. From the graphical results in previous sections, it is clear that both m and hm consistently result in clustered outcomes, while eg obtains a wider spread across the objective space, even though outcomes in m and hm may be more desirable from the tracking performance point of view. A point to note here is that in the case of eg, the IGD indicator is being largely influenced by the evenness in spread of outcomes with respect to the reference set, which is indeed qualified by low $IGDV$ values for eg. As can be seen, the large number of "+" symbols show eg consistently outperforming m and hm in $IGDV$. The IGD values for sm suggest it to be competitive across scenarios

Table V. Medians over 30 independent runs of the IGD and $IGDV$ indicators for Pareto frontiers pertaining to only homogeneously mixed (m), only heterogeneously mixed (hm), only EPSILON GREEDY (eg), only SOFTMAX (sm), and only UCB (ucb) outcomes, respectively.

	Mixed fronts			Learnt fronts					
	IGD								
Scenario	m	hm		eg		sm		ucb	
1	0.2976	0.2945	=	0.4730	−−	0.2552	++	0.2916	==
2	0.3632	0.3395	+	0.2615	++	0.3641	=−	0.3238	+=
3	0.3595	0.3148	+	0.2643	++	0.3641	=−	0.3521	=−
4	0.3348	0.3406	=	0.2615	++	0.3409	==	0.3454	==
5	0.3717	0.3943	−	0.2744	++	0.3525	++	0.3811	=+
6	0.3530	0.3665	=	0.3335	++	0.3687	==	0.3851	−−
7	0.3549	0.3753	−	0.2739	++	0.3568	=+	0.3596	=+
8	0.3502	0.3596	=	0.2792	++	0.3595	==	0.3652	==
9	0.3640	0.3776	−	0.2994	++	0.3593	=+	0.4034	−−
10	0.3844	0.4224	−	0.3450	==	0.4114	−+	0.3998	=+
11	0.3541	0.3425	=	0.3908	−−	0.3858	−−	0.3950	−−
12	0.3768	0.4031	−	0.3191	++	0.4494	−−	0.3758	=+
	$IGDV$								
	m	hm		eg		sm		ucb	
1	0.0604	0.0422	+	0.0666	=−	0.0188	++	0.0166	++
2	0.0706	0.0452	+	0.0439	+=	0.0495	+=	0.0338	++
3	0.0759	0.0459	+	0.0334	++	0.0535	+=	0.0377	+=
4	0.0672	0.0547	+	0.0239	++	0.0409	++	0.0341	++
5	0.0792	0.0645	+	0.0293	++	0.0611	+=	0.0550	++
6	0.0600	0.0622	=	0.0630	==	0.0635	==	0.0629	==
7	0.0578	0.0525	+	0.0273	++	0.0449	++	0.0417	++
8	0.0531	0.0587	=	0.0310	++	0.0559	==	0.0492	++
9	0.0705	0.0763	−	0.0290	++	0.0619	++	0.0604	++
10	0.0673	0.0735	−	0.0556	=+	0.0718	=+	0.0644	=+
11	0.0640	0.0427	+	0.0551	=−	0.0535	+−	0.0523	+−
12	0.0861	0.0767	+	0.0545	++	0.0723	++	0.0571	++

Note: The Wilcoxon rank sum test was used with a 95% confidence level to assess statistical significance. The following symbols denote significant difference in IGD and IGDV of a front w.r.t mixed outcome fronts m or hm: "+" indicates a significant difference in favour of the front, "−" indicates a significant difference in favour of the mixed outcome front being compared against, "=" indicates similarity. The ordering of the symbols pertains to the comparison being made against front m, followed by front hm.

w.r.t. m and hm. This is indicated by the large number of "=" symbols for sm, and an equal number of "+" and "−" symbols. Although sm may come as close to the reference set as m and hm across scenarios, it is consistently more evenly spread than m or hm. Finally, the IGD values of the ucb frontiers are competitive compared with those for m and hm. But again, the $IGDV$ values suggest better spreads along the reference set, arising from the learnt strategy.

9. CONCLUSIONS

We have studied the self-organising behaviour of smart camera networks which use auctions to exchange object tracking responsibilities during runtime. Our first contribution was to show that heterogeneous configurations of marketing strategies in the network can lead to increased network-level tracking performance while simultaneously decreasing the number of auction invitations, a proxy for communication and processing overhead. I.e. heterogeneity led to more Pareto efficient outcomes than those possible in homogeneous configurations. We demonstrated this on a range of scenarios, both using an open source simulation package and real video feeds.

Our second contribution is to show that dynamic heterogeneous configurations, when cameras change their strategies during runtime, give rise to high Pareto efficiency, typically comparable with the most efficient of the outcomes from the static heterogeneous case. We studied two approaches to the generation of dynamic hetero-

geneity. Firstly, we used a non-adaptive technique based on the idea of mixed strategies from game theory, which uses stationary probability distributions to generate behaviours at run time. Secondly, we used an adaptive technique, which uses bandit solving algorithms to learn behaviours online, in a decentralised fashion within each camera, based on that camera's feedback from the environment. Both of these approaches were able to provide increased Pareto efficiency relative to the homogeneous case.

Our third contribution is to show that, in some cases, outcomes arising from online learning extend the Pareto frontier even when compared to the best possible outcomes from an exhaustive analysis of possible static heterogeneous configurations. Online learning can not only enable a network of cameras to find high performing heterogeneous configurations, but in some scenarios is able to obtain favourable outcomes which cannot be reached in the static case.

Our fourth contribution is to show that outcomes arising from online learning were typically more evenly spread across the objective space than those arising from mixed or heterogeneously mixed strategies. Run time feedback in the adaptive case enabled the addition of firstly normalisation by distribution and secondly a weighting parameter in the reward function to help achieve this.

Although the results in this paper are obtained from a camera network case study, the principles behind both heterogeneous configuration, dynamic configuration and decentralised online learning are not limited to camera networks. Indeed, the results in this article provide insight into heterogeneity, dynamism of behaviour and adaptivity more generally in decentralised networked systems. Firstly, by increasing the number of possible configurations, heterogeneity offers more possibilities for network configuration. In all the cases studied in this article, some of these new possibilities led to increased Pareto efficiency of the network, relative to homogeneous configurations, while some led to decreased Pareto efficiency. Deployed correctly, heterogeneous configurations have a key positive influence on Pareto efficiency. Dynamic behaviour over time offers yet more configuration possibilities, however the potential increase in Pareto efficiency by the addition of dynamism *per se* was not so great as in the case of heterogeneity relative to homogeneity. Perhaps surprisingly however, outcomes arising from both mixed strategy based dynamic behaviour and dynamic behaviour arising from online learning outperformed the majority of static heterogeneous configurations and virtually all homogeneous configurations on all the scenarios studied. This suggests that in addition to heterogeneity, dynamic forms of heterogeneity are still in themselves useful. While the Pareto efficient frontiers arising from static heterogeneity were usually the highest, the most consistently Pareto efficient outcomes were obtained by those networks which used online learning to adaptively select behaviours based on environmental feedback, captured in the form of a reward function. Of the online learning techniques evaluated, SOFTMAX typically obtained the most efficient results.

A general problem which then arises in the deployment of decentralised systems is how to make use of such heterogeneity and dynamism. Though the Pareto frontiers arising from static heterogeneity were typically the most efficient, one cannot simply deploy *static heterogeneity*. Rather this is a characterisation of a vast space of configuration options; selecting the correct static heterogeneous configuration relies on advance scenario knowledge and calibration, the absence of which was a key motivator for the overall approach being studied in this article. Instead, the online learning approaches represent a far more realistic deployment option: firstly no advance scenario knowledge is needed, since behaviour is learnt through environmental interaction, and secondly the presence of a feedback loop enables a parameter to be used in

order to favour regions of the Pareto frontier according to the operator's preferences. We encapsulated this in the form of a value α in the learning reward function.

There are many avenues for extending this line of research. Firstly, further evaluation of the approaches introduced here using more complex deployments may provide insight into when different bandit solvers outperform each other, and when they outperform static configurations. Secondly, in this paper we have focussed our study on classic, well known bandit solvers. There are many other bandit solver algorithms to be found in the literature, some of which are designed explicitly with unbounded rewards and dynamic reward distributions in mind. These, along with other online learning techniques, may provide further improvements.

As an important aspect of this, due to complex interactions between trading, vision graph learning and bandit solvers, the local-global mapping assumed in the local reward function used by the online learning methods could be further tuned. Though the results presented in this paper based on linear scalarisation are encouraging, and results published elsewhere [Moffaert et al. 2014] suggest that this scalarisation method is appropriate, it is likely that improved spread in particular would be achieved with more advanced techniques. Fundamentally, how to define multi-attribute reward functions at the local level, such as to achieve desired Pareto efficient outcomes at the global level, remains an open question.

Furthermore, as briefly alluded to in Section 5.3, one potential alternative avenue of research is to perform a coevolutionary analysis of the mixed strategies. Coevolutionary algorithms have been successfully applied to complex multi-agent learning problems from the idealised Iterated Prisoner's Dilemma [Axelrod 1987] to more complex economic games (e.g. [Phelps et al. 2008; Lewis et al. 2010]). Such an analysis has the potential here to both identify high performing complementary mixed strategies, as well as provide insight into the structure of the strategy search space.

Finally, from the perspective of the smart camera application, we made two simplifying assumptions in order to aid our analysis. Firstly, we assumed in this work that we are able to recognise and track an object correctly in all cases. However, as discussed e.g. by Nebehay et al. [2013], this is in itself not a trivial task. Secondly, we assumed that the number of objects currently tracked by a camera does not affect its tracking performance. However, the need for tracking algorithms to share finite resources on a camera means that this will likely not be the case when the number of objects is sufficiently large. Future smart camera research should therefore address the question of the robustness of the results in this work, with respect to a degradation of tracking success rate.

REFERENCES

Gerrit Anders, Christian Hinrichs, Florian Siefert, Pascal Behrmann, Wolfgang Reif, and Michael Sonnenschein. 2012. On the Influence of Inter-Agent Variation on Multi-Agent Algorithms Solving a Dynamic Task Allocation Problem under Uncertainty. In *Proceedings of the Sixth IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2012)*. IEEE Computer Society Press, Washington, DC, USA, 29–38.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. 2002. Finite-time Analysis of the Multiarmed Bandit Problem. *Machine Learning* 47 (2002), 235–256.

Robert Axelrod. 1987. The evolution of strategies in the Iterated Prisoners Dilemma. In *Genetic algorithms and simulated annealing*, LLawrence Davis (Ed.). Pittman, London, 32–41.

Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. 2008. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding* 110, 3 (2008), 346–359.

Ken Binmore. 2007. *Game Theory: A Very Short Introduction*. Oxford University Press.

Adam Campbell, Cortney Riggs, and Annie S. Wu. 2011. On the Impact of Variation on Self-Organizing Systems. In *Proceedings of the Fifth IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2011)*. IEEE Computer Society Press, Washington, DC, USA, 119–128.

Zhaolin Cheng, Dhanya Devarajan, and Richard J. Radke. 2007. Determining Vision Graphs for Distributed Camera Networks using Feature Digests. *EURASIP J. Appl. Signal Process.* 2007, 1 (Jan. 2007), 220–220.

Carlos A. Coello Coello and Nareli Cruz Cortés. 2005. Solving Multiobjective Optimization Problems Using an Artificial Immune System. *Genetic Programming and Evolvable Machines* 6, 2 (2005), 163–190.

Henry Detmold, Anton van den Hengel, Anthony Dick, Alex Cichowski, Rhys Hill, Ekim Kocadag, Katrina Falkner, and David S. Munro. 2007. Topology Estimation for Thousand-Camera Surveillance Networks. In *Proceedings of the ACM/IEEE International Conference on Distributed Smart Cameras*. Vienna, Austria, 195–202.

Giovanna Di Marzo Serugendo, Marie-Pierre Gleizes, and Anthony Karageorgos (Eds.). 2011. *Self-organising Software: From natural to artificial adaptation*. Springer.

Bernhard Dieber, Christian Micheloni, and Bernhard Rinner. 2011. Resource-Aware Coverage and Task Assignment in Visual Sensor Networks. *IEEE Transactions on Circuits and Systems for Video Technology* 21, 10 (2011), 1424–1437.

Ugur Murat Erdem and Stan Sclaroff. 2005. Look there! Predicting where to look for motion in an active camera network. In *Proceedings of the IEEE Conference on Vision and Signal-based Surveillance*. Como, Italy, 105–110.

Lukas Esterle, Peter R. Lewis, Bernhard Rinner, and Xin Yao. 2012. Improved Adaptivity and Robustness in Decentralised Multi-Camera Networks. In *Proceedings of the Sixth ACM/IEEE International Conference on Distributed Smart Cameras*. IEEE Press, 1–6.

Lukas Esterle, Peter R. Lewis, Xin Yao, and Bernhard Rinner. 2014. Socio-Economic Vision Graph Generation and Handover in Distributed Smart Camera Networks. *ACM Transactions on Sensor Networks* 10, 2, Article 20 (2014).

Matteo Gagliolo and Jürgen Schmidhuber. 2011. Algorithm Portfolio Selection as a Bandit Problem with Unbounded Losses. *Annals of Mathematics and Artificial Intelligence* 61, 2 (2011), 49–86.

Omar Javed, Zeeshan Rasheed, Khurram Shafique, and Mubarak Shah. 2003. Tracking Across Multiple Cameras With Disjoint Views. In *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2 (ICCV '03)*. IEEE Computer Society, Washington, DC, USA, 952–.

Peter R. Lewis, Lukas Esterle, Arjun Chandra, Bernhard Rinner, and Xin Yao. 2013. Learning to be Different: Heterogeneity and Efficiency in Distributed Smart Camera Networks. In *Proceedings of the 7th IEEE Conference on Self-Adaptive and Self-Organizing Systems (SASO)*. IEEE Press, 209–218.

Peter R. Lewis, Paul Marrow, and Xin Yao. 2010. Resource Allocation in Decentralised Computational Systems: An Evolutionary Market Based Approach. *Autonomous Agents and Multi-Agent Systems* 21, 2 (2010), 143–171.

Yiming Li and Bir Bhanu. 2009. A Comparison of Techniques for Camera Selection and Handoff in a Video Network. In *Proceedings of the ACM/IEEE International Conference on Distributed Smart Cameras*. Como, Italy, 1–8.

Yiming Li and Bir Bhanu. 2011. Utility-based Camera Assignment in a Video Network: A Game Theoretic Framework. *IEEE Sensors Journal* 11, 3 (2011), 676–687.

Dimitrios Makris, Tim Ellis, and James Black. 2004. Bridging the Gaps between Cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 205–210.

Zehavit Mandel, Ilan Shimshoni, and Daniel Keren. 2007. Multi-Camera Topology Recovery from Coherent Motion. In *Proceedings of the ACM/IEEE International Conference on Distributed Smart Cameras*. Vienna, Austria, 243–250.

Kristof Van Moffaert, Tim Brys, Arjun Chandra, Lukas Esterle, Peter R. Lewis, and Ann Nowé. 2014. A Novel Adaptive Weight Selection Algorithm for Multi-Objective Multi-Agent Reinforcement Learning. In *Proceedings of the Annual International Joint Conference on Neural Networks (IJCNN) 2014*. IEEE Press.

Birgit Möller, Thomas Plötz, and Gernot A. Fink. 2008. Calibration-free Camera Hand-Over for Fast and Reliable Person Tracking in Multi-Camera Setups. In *Proceedings of the 19th International Conference on Pattern Recognition*. Tampla, FL, USA, 1–4.

Kazuyuki Morioka, Szilveszter Kovacs, Joo-Ho Lee, and Peter Korondi. 2010. A Cooperative Object Tracking System with Fuzzy-Based Adaptive Camera Selection. *International Journal on Smart Sensing and Intelligent Control* 3, 3 (2010), 338–358.

Eduardo F. Nakamura, Heitor S. Ramos, Leandro A. Villas, Horacio A.B.F. de Oliveira, Andre L.L. de Aquino, and Antonio A.F. Loureiro. 2009. A reactive role assignment for data routing in event-based wireless sensor networks. *Computer Networks* 53, 12 (2009), 1980–1996.

Georg Nebehay, Walter Chibamu, Peter R. Lewis, Arjun Chandra, Roman Pflugfelder, and Xin Yao. 2013. Can Diversity amongst Learners Improve Online Object Tracking? In *Multiple Classifier Systems*, Zhi-

Hua Zhou, Fabio Roli, and Josef Kittler (Eds.). Lecture Notes in Computer Science, Vol. 7872. Springer Berlin Heidelberg, 212–223.

Steve Phelps, Kai Cai, Peter McBurney, Jinzhong Niu, Simon Parsons, and Elizabeth Sklar. 2008. Auctions, evolution, and multi-agent learning. In *Adaptive Agents and Multi-Agent Systems III. Adaptation and Multi-Agent Learning (Lecture Notes in Computer Science)*, Vol. 4865. Springer, 188–210.

Arun Prasath, Abhinay Venuturumilli, Aravind Ranganathan, and Ali A. Minai. 2009. Self-Organization of Sensor Networks with Heterogeneous Connectivity. In *Sensor Networks: Where Theory Meets Practice*, Gianluigi Ferrari (Ed.). Springer, 39–59.

Markus Quaritsch, Markus Kreuzthaler, Bernhard Rinner, Horst Bischof, and Bernhard Strobl. 2007. Autonomous Multicamera Tracking on Embedded Smart Cameras. *EURASIP Journal on Embedded Systems Volume 2007* (2007), 10.

Faisal Qureshi and Demetri Terzopoulos. 2008. Multi-camera Control through Constraint Satisfaction for Persistent Surveillance. In *IEEE Conference on Vision and Signal-based Surveillance*. Santa Fe, USA, 211–218.

Dominik Rojković, Tihomir Crnic, and Igor Cavrak. 2012. Agent-based topology control for wireless sensor network applications. In *MIPRO, 2012 Proceedings of the 35th International Convention*. MIPRO, Rijeka, Croatia, 277–282.

Kay Römer, Christian Frank, Pedro José Marrón, and Christian Becker. 2004. Generic role assignment for wireless sensor networks. In *Proceedings of the 11th ACM SIGOPS European Workshop*. ACM Press, New York, NY, USA.

Norman Salazar, Juan A. Rodriguez-Aguilar, and Josep Lluis Arcos. 2010. Self-Configuring Sensors for Uncharted Environments. In *Proceedings of the Fourth IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2010)*. IEEE Computer Society Press, Washington, DC, USA, 134–143.

Richard S. Sutton and Andrew G. Barto. 1998. *Reinforcement Learning: An Introduction*. MIT Press.

Peter Vamplew, Richard Dazeley, Adam Berry, Rustam Issabekov, and Evan Dekker. 2011. Empirical Evaluation Methods for Multiobjective Reinforcement Learning Algorithms. *Machine Learning* 84, 1-2 (2011), 51–80.

Peter Vamplew, John Yearwood, Richard Dazeley, and Adam Berry. 2008. On the Limitations of Scalarisation for Multi-objective Reinforcement Learning of Pareto Fronts. In *Proceedings of the 21st Australasian Joint Conference on Artificial Intelligence: Advances in Artificial Intelligence (AI '08)*. Springer-Verlag, Berlin, Heidelberg, 372–378.

Kristof Van Moffaert, Madalina M. Drugan, and Ann Nowé. 2013a. Hypervolume-Based Multi-Objective Reinforcement Learning. In *Evolutionary Multi-Criterion Optimization*, Robin C. Purshouse, Peter J. Fleming, Carlos M. Fonseca, Salvatore Greco, and Jane Shaw (Eds.). Lecture Notes in Computer Science, Vol. 7811. Springer, 352–366.

Kristof Van Moffaert, Madalina M. Drugan, and Ann Nowé. 2013b. Scalarized multi-objective reinforcement learning: Novel design techniques. In *IEEE Symposium on Adaptive Dynamic Programming And Reinforcement Learning (ADPRL)*. IEEE Press, 191–199.

Lyndon While, Phil Hingston, Luigi Barone, and Simon Huband. 2006. A faster algorithm for calculating hypervolume. *IEEE Transactions on Evolutionary Computation* 10, 1 (2006), 29–38.

Michael J. Wooldridge. 2001. *Introduction to Multiagent Systems*. John Wiley and Sons Inc.

Qingfu Zhang, Aimin Zhou, and Yaochu Jin. 2008. RM-MEDA: A Regularity Model-Based Multiobjective Estimation of Distribution Algorithm. *Evolutionary Computation, IEEE Transactions on* 12, 1 (Feb 2008), 41–63.

Received Sometime in 2013; revised Sometime in 2014; accepted Sometime in 2014