Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

Decentralized Predictive Sensor Allocation

Mark Ebden*, Mark Briers**, and Stephen Roberts*

Abstract— **We present a method of dynamic coalition formation (DCF) in sensor networks to achieve well-informed sensor-target allocations. Forecasts of target movements are incorporated when choosing sensor states, as is a memory of target observation. The algorithm can be run in a centralized or decentralized configuration; the latter relies on local message passing in the form of the max-sum algorithm. We show how the DCF algorithm has been applied to synthetic and real data.**

I. Introduction

Our work contributes to a sensor-management topic known as *sensor resource allocation* (synonyms include "sensor behaviour assignment", "sensor action planning", "sensor selection", and "sensor-to-task assignment" [8], [12]). Although this topic dates to at least the 1970s [11], little has been published on decentralized dynamic allocation. We propose a general framework; the results have implications outside the sensing community, as sensor networks are merely one of the classic applications of *multi-agent systems* [15]. Other applications that might benefit from our approach are aircraft-crew scheduling and disaster management (including emergency medical response planning), and problems with similarly complex dynamics.

In such problems, approaches more exhaustive than the "greedy" allocation algorithms preferred in simple search spaces may be called for, in order to avoid overlooking optimal solutions. One of the chief drawbacks of these approaches, however, is that when the required temporal resolution of a forecast (over which the environment is predicted and a system's response simulated) is too fine, a combinatorial explosion occurs: if approximately G actions are available to a system at each of W time steps, the number of unique action sequences to consider is approximately $\mathcal{O}(G^W)$. To solve this issue, we consider the paradigm (e.g. see Konishi and Ray [7]) of weighing immediate coalitional changes against delayed coalitional changes:

> "A process of coalition formation is an equilibrium if at any date and at any going state, a coalitional move to some other state can be 'justified' by the very same scheme applied in future: the coalition that moves must have higher present value (starting from the state it moves *to*) for each of its members, compared to (one-period) inaction under the going state." [7]

* M. Ebden and S. Roberts are with the Department of Engineering Science, University of Oxford, Oxford, OX1 3PJ, United Kingdom. mark.ebden@eng.ox.ac.uk Phone: +44-1865-283391.

** M. Briers is with QinetiQ Ltd, Malvern Technology Centre, St. Andrews Road, Malvern, WR14 3PS, United Kingdom.

Thus we have adapted a principle of dynamic coalition formation (DCF) to the case of finding optimal sensor allocations. We describe a general solution method for other environments with complex dynamics.

The sensors in the specific problem discussed in this paper represent a possibly heterogenous mix of infrared cameras, video cameras, multispectral imaging, radar, and other technology. The sensors are tasked with observing moving objects, or *targets*. Ordinarily, three aspects of interest are target position, velocity, and identity classification (where the identity may be one of vehicle, trespasser, etc.), but for simplicity here we assume the first two are given: i.e. a surveillance sensor such as a large-scale surface-movement radar provides suitably accurate positions and velocities.

Most sensors have a limited field of view (usually $<180°$) at any one time; therefore, if a sensor can rotate, its orientation ought to be coordinated with those of its neighbours. Field of view and orientation are the two state variables considered in this work, but these are easily adjustable to model other systems.

If an allocation algorithm is run periodically, the assignments will gradually change as targets move. Without forecasting, this process is simply a non-dynamic algorithm being run many times. Our aim is to consider algorithms which forecast target movements when determining the current allocations, and which take into account the time delays associated with the state changes of sensors. In this way, we can coordinate time-dependent coalitional activities.

As a simple example, consider the three sensors in Fig. 1a, observing four targets. Imagine that a surveillance sensor (not shown) identifies a new target (α) approaching rapidly from the east. A non-dynamic algorithm might try to reallocate the sensors based largely on sensor-target proximity (see Fig. 1b). However, during the time that Sensors 2 and 3 are rotating, two targets are not being observed. Thus, a good dynamic algorithm might instead allocate Sensor 1 to α, while preparing Sensor 3 for a future reallocation by adjusting its angle slightly (see Fig. 1c). The DCF algorithm which we have developed is capable of yielding this type of improvement, and other benefits of dynamic coordination. The sensor-network framework we consider is not a constraint on our approach; it is merely used to place the algorithm's operations into a meaningful context.

We will begin by describing a standard message-passing approach through which the sensors communicate. Section III describes the DCF problem they are to solve, and our solution for the case of simple forecasts of the environment (only one time point in the future). Section IV outlines our method of solving the problem for arbitrarily lengthy and

978-1-4244-3124-3/08/$25.00 ©2008 IEEE

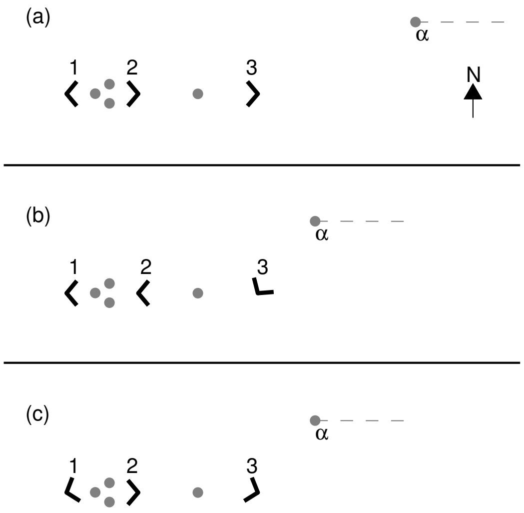

Fig. 1. An example of how a dynamic algorithm can react with foresight to a changing coalition formation environment. (a) Grey circles are targets to observe, and the three black chevrons are sensors with the capacity to rotate slowly. Target α has recently appeared. (b) A non-dynamic algorithm might aim to assign Sensor 3 to α. (c) A dynamic algorithm might instead assign Sensor 1 to α, and prepare Sensor 3 for a future reassignment.

complex forecasts. The method is validated in Section V with examples, and Section VI closes with a discussion.

II. Message Passing

Our decentralized approach relies on local message passing, a technique which has been used successfully in inference problems [13] but to our knowledge has not yet been applied to the problem of dynamic sensor allocation. Information is shared locally among agents in an effort to solve a global problem [10]. Here, this problem is to discover, in a sensor network, which states the sensors should assume in order to maximize the expected *number of targets identified* in a given upcoming time window; however, a similar framework could be used on any multi-agent system with tasks to be assigned.

We have chosen to use message passing to decentralize the problem chiefly because large sensor networks tend to burden the allocation algorithm with a combinatorial explosion. The message-passing technique ensures that the amount of information to compute remains manageable for even the vastest networks, provided the connectedness is realistically limited — e.g. sensors on opposite sides of the network cannot share targets.

Our work is based on factor graphs and an extension of the sum-product algorithm (also known as belief propagation; see [9] for a tutorial) called the max-sum algorithm. Farinelli *et al.* [4] pioneered the application of this algorithm in multi-agent coordination. The authors showed that their method works well in graph-colouring problems and implemented it in hardware successfully using wireless embedded devices. The method is extensible to many different types of distributed problems.

In our problem, sensors in a network are deemed to be *neighbours* when they have the ability to bring at least one common target within range — even if the sensors are not currently allocated to it at a given moment. The state of the nth sensor is described by a *variable*, x_n, which acts as an index pointing to a complete state description. The set of states in the mth sensor's neighbourhood (including itself) at any one time is $\mathbf{x}_m$. (For example, in a three-sensor network, it may be that $\mathbf{x}_1 = \{x_1, x_2\}$. Finally, the utility of each sensor is described by a *factor*, $U(\mathbf{x}_m)$. For example, $U(\mathbf{x}_1)$ might indicate that the first of two sensors is expected to identify a total of three targets in the upcoming time window; non-integer estimates are also permitted. When several sensors point at the same target, the utility is divided equally. Our notation for partitioning each sensor into one variable and one factor is broadly consistent with that of Farinelli *et al.* [4]. Our method of selecting candidate states x_n and of calculating the factors $U(\mathbf{x}_m)$ is described in the next two sections.

Each variable in the network communicates bidirectionally with its associated factor as well as the factors in neighbouring sensors, until the entire network converges. A simplistic description is that, before evaluating a possible coalition switch, a sensor receives a report from each of its neighbours on the expected ramifications in the neighbours' neighbourhoods; these reports are propagated in the form of Q and R messages.

More specifically, the set of factors connected to the nth variable is referred to as $\mathcal{M}(n)$, and the set of variables connected to the mth factor as $\mathcal{N}(m)$. A message Q sent from the nth variable to the mth factor is composed as follows:

$$Q_{n\to m}(x_n) = \alpha_{nm} + \sum_{m' \in M(n)\setminus m} R_{m'\to n}(x_n), \qquad (1)$$

where $R_{m'\to n}(x_n) = 0$ for the first iteration, and the scalar α_{nm}, initially zero, is updated after each iteration such that

$$\sum_{x_n} Q_{n\to m}(x_n) = 0. \qquad (2)$$

In return, messages from each factor to all of its neighbouring variables are then reciprocated. The message R from the mth factor to the nth variable is composed as follows:

$$R_{m\to n}(x_n) = \max_{\{x_{n'}: n' \in \mathcal{N}(m)\setminus n\}} \Bigg[U(\mathbf{x}_m) + \sum_{n'' \in \mathcal{N}(m)\setminus n} Q_{n''\to m}(x_{n''}) \Bigg]. \qquad (3)$$

It can be shown [4] that after convergence, the network's total utility or *system welfare* $V = \sum U(\mathbf{x}_m)$ is maximized approximately when each sensor assumes its optimal state x_n^* given by

$$x_n^* = \underset{x_n}{\operatorname{argmax}} \Bigg[\sum_{m \in \mathcal{M}(n)} R_{m\to n}(x_n) \Bigg]. \qquad (4)$$

The maximized utility is guaranteed to be optimal only when the sensors are connected as an “acyclic graph”, e.g. as a tree.

We emphasize that the dynamic coalition formation algorithm can be applied in a centralized or decentralized configuration; the use of local message passing is an attractive option primarily for problems which scale poorly with the number of sensors/agents.

III. Dynamic Coalition Formation

To represent the problem of allocating I sensors to J targets, we build on the coalition-formation notation of Dang *et al.* [2] and Fernández *et al.* [5]. At each discrete time step τ, the ith sensor in $\mathcal{I} = \{1, 2, \ldots, i, \ldots, I\}$ can assume any of a number of states x_i to observe one or more of the targets $\mathcal{T} = \{t_1, t_2, \ldots, t_j, \ldots, t_J\}$. In doing so it incurs a cost $c_i(\tau)$, which ordinarily would reflect anything from electrical power to a penalty for assuming useless states, and here $c_i(\tau)$ reflects only the time lost when changing from one state to another. All sensors observing target t_j belong to the coalition $C_j(\tau) \subseteq \mathcal{I}$, which has a *value*, $v(C_j(\tau), t_j)$, indicating how useful it is to observe a particular target with a particular coalition. The set of all J coalitions existing at time τ is the *coalition structure*, $\text{CS} = \{C_1(\tau), \ldots, C_J(\tau)\}$.

System welfare, introduced in the previous section, is calculated as

$$V(\tau) = \sum_{t_j \in \mathcal{T}} v\big(C_j(\tau), t_j\big) - \sum_{i \in \mathcal{I}} c_i(\tau). \tag{5}$$

The optimal coalition structure, CS^*, is then $\underset{\text{CS}\in\Gamma(\mathcal{I},\mathcal{T})}{\operatorname{argmax}} V(\tau)$, where $\Gamma(\mathcal{I}, \mathcal{T})$ is the set of all possible coalition structures.

At least two enhancements to this model should be made for dynamic environments. First, sensors require time to be reallocated, and our approach is to affect $c_i(\tau)$ by considering the impact of the state transition time. Second, targets which have been previously identified may be of lesser interest to observe than unidentified targets. This will be addressed by making $v\big(C_j(\tau), t_j\big)$ dependent on events which occurred in the past; i.e. the value of a coalition is influenced by what information has been gleaned previously from the environment.

A. Coalition Values

On a two-dimensional grid, the changing locations and velocities of the J possibly heterogeneous targets $\mathcal{T}$ introduced above are gradually made known at each time τ. (These locations might be determined by a surveillance sensor, as noted in the Introduction.) Recall that the I sensors $\mathcal{I}$ observe the targets to gradually *determine their identities*. For identification to occur, a target must lie within a sensor's (adjustable) sector-shaped area of observation. At each time step, a sensor faces one of an infinite number of orientations, and assumes one of an infinite number of fields of view (affecting the degree of zoom, and represented here as the central angle of the sensor's observation sector), both measured in radians.

If the jth target is moving with velocity $v_j(\tau)$, and is separated from the ith sensor by distance $d_{ij}(\tau)$, and the sensor has field of view $z_i(\tau)$ (measured in radians), then, provided the target lies somewhere within the sensor's observation sector, the chance that the target may be identified within the current time step is expressed as

$$\begin{aligned} p_{ij}(\tau) \quad = \quad & p_{\max}\, \Omega\big(|v_j(\tau)|, v_{\max}\big)\, \Omega\big(d_{ij}(\tau), d_{\max}\big) \cdot \\ & \Omega\big(z_i(\tau), z_{i\max} + \text{constant}\big) \end{aligned} \tag{6}$$

where

$$\Omega(a, a_{\max}) = \sup\left\{0, \frac{2}{1 + \frac{a}{a_{\max}}} - 1\right\}. \tag{7}$$

Hence the likelihood of a successful identification decreases with the speed of a target, the sensor-target distance, and the field of view of the sensor. The meanings of the parameters $v_{\max}$, $d_{\max}$, and $z_{i\max}$ are, respectively, the speed at which a target becomes an unidentifiable blur, the distance at which a target becomes an unidentifiable speck, and the widest possible field of view a sensor can assume. $p_{\max}$ is the highest possible probability of identification, which occurs when the target velocity and separation are zero and the sensor has zoomed maximally.

The value of a coalition, i.e. the overall probability of identifying the jth target without consideration of costs, takes into account simultaneously but independently all sensors in its coalition:

$$v\big(C_j(\tau)\big) = 1 - \prod_{i \in C_j(\tau)} \left[1 - p_{ij}(\tau)\right]. \tag{8}$$

The fairly basic scheme which we have used to describe the general problem is easy to modify for a given actual scenario. In particular, rather than treating each sensor's contribution independently, more sophisticated target identification probability distributions may be used, e.g. distributions that consider synergy among coalition members. Independence among the time steps is also easy to remove, e.g. by considering the duration of observation when calculating $v\big(C_j(\tau), t_j\big)$.

B. Coalition Costs

During the time that the ith sensor moves from some state $x_i = a$ to another state $x_i = b$ it is inactive for a certain number of time units, $e_{(a\to b)_i}$. Sensors can follow slow targets without penalty, i.e. a certain amount of sensor rotation is permitted during each time step without affecting the probabilities of target identification. The total number of time units for which the sensor is inactive while moving from state a to state b is

$$e_{(a\to b)_i} = \sup\left\{\Delta_{\text{F}}|F_a - F_b|,\ \Delta_\theta \theta_{ab} - \Delta_0\right\} \tag{9}$$

where $\Delta_{\text{F}}, \Delta_\theta \geq 0$ are the per-radian penalties for field-of-view and orientation adjustments, respectively; F_a and F_b are fields of view; $0 \leq \theta_{ab} \leq 180°$ is the non-reflex angle between the two sensor orientations; and $\Delta_0 \geq 0$ is the angle (in radians) which a sensor can rotate without penalty at each time step.

In this manner, we have generalized the costs $c_i(\tau)$ to reflect not only coalition change (e.g. switching to observing a new target would often be associated with $e_{(a\to b)_i} \neq 0$) but also the ongoing costs of maintaining a coalition (e.g. if a target is moving rapidly). A cost should reflect the actual reduction in the probability of target identification; we have shown elsewhere [3] that, for $E_i(\tau) < 1$, the function which maximizes target identification in a single time step is

$$c_i(\tau) = \sum_{t_j \in \mathcal{T}} E_i(\tau) p_{ij}(\tau) \prod_{h \neq i} \big\{1 - p_{ij}(\tau)\big[1 - E_h(\tau)\big]\big\} \tag{10}$$

where $E_i(\tau)$ is $e_{(a\to b)_i}$ at time τ. This cost formula guarantees that the system welfare, given in Eq. (5), is equal to the expected number of targets which will be identified in a particular time step, *after* considering the effects of sensor movement. Eq. (10) only addresses a single time step, and for $E_i(\tau) \gtrsim 1$, naturally it becomes useful to take advantage of longer forecasts when evaluating the system welfare. This approach is the subject of the next section.

IV. Long-term Dynamic Coalition Formation

Current positions and velocities are revealed to all sensors at each time step τ, from which future values can be extrapolated. To handle the diversity of possible target dynamics, a general tracking framework such as VS-IMM or JMLS [1] would ordinarily be preferred; however, for simplicity we currently employ a constant-velocity model to form these projections. To account for the uncertainty, the expression for $p_{ij}(\tau)$ is marginalized over the various probabilities in the prediction, such as whether the target will fall in the sensor's field of view. A key feature of our work is that extension of this time horizon arbitrarily into the future does not cause a combinatorial explosion.

If the ith sensor is surrounded by $J_i(\tau)$ within-range targets (not necessarily visible simultaneously), and the sensor's field of view and orientation are adjustable with infinitesimal resolution, then the number of possible different *collections* (groupings) of the $J_i(\tau)$ targets is at most $\frac{1}{2}J_i(\tau)[J_i(\tau)+1]$. (In other problems this expression is a factorial "combination", but here it is simplified by geometrical findings.) Hence, among sensors $\mathcal{I}$, a time window of width W units corresponds to a search space of maximum size

$$\frac{1}{2}\prod_{\tau=1}^{W}\left\{\prod_{i\in\mathcal{I}}\left[J_i(\tau)^2 + J_i(\tau)\right]\right\}, \tag{11}$$

which is too vast to search exhaustively when W is large. Our approach considers two time windows, of length W_1 and W_2. The first is applied to rapidly find which sensor state combinations $s_q \in \mathbf{s}$ might be useful, and the second leverages the delay paradigm from the Introduction to estimate the effectiveness of each of those combinations s_q, by counting the number of target identifications expected if a "greedy" algorithm were used for each of the $W_2 - 1$ remaining time steps.

The method is applied to each neighbourhood of sensors, including only the targets applicable to that neighbourhood during W_1. The method, designed to not increase exponentially in complexity with the length of the time horizon, is as follows:

1) The first window size is set to

$$W_1 = \left\lceil \sup_{i,b}\ e_{(a\to b)_i} \right\rceil. \tag{12}$$

Note that maximal $e_{(a\to b)_i}$ corresponds to a rotation of π radians or an extreme change in the field of view, whichever takes longer.

2) The projected positions of the targets throughout this window are assessed and standard deviations assigned, using a constant-velocity model.

3) At each time step $\tau = \{1, 2, \ldots, W_1\}$, and for each sensor i: a) A list $\mathcal{T}_i(\tau) = \{\mathbf{t}_{i1}(\tau), \mathbf{t}_{i2}(\tau), \mathbf{t}_{i3}(\tau), \ldots\}$ of the $J_i(\tau)$ possible target collections visible to the sensor is compiled by considering which targets can be viewed simultaneously. Strictly speaking, nonzero *potential* for viewing is considered, since the projected target positions are only estimates.
b) For each collection $\mathbf{t}_{ih}(\tau) \in \mathcal{T}_i(\tau)$, only the sensor states $x_{ih}(\tau)$ which might allow the observation of this collection are retained. Of these, two optimal sensor states are isolated:

$$x_{ip}(\tau) = \operatorname*{argmax}_{x_{ih}(\tau)} \left\{ \sum_{t_j \in \mathbf{t}_{ih}(\tau)} p_{ij}(\tau) \right\} \tag{13}$$

$$x_{id}(\tau) = \operatorname*{argmax}_{x_{ih}(\tau)} \left\{ \sum_{t_j \in \mathbf{t}_{ih}(\tau)} p_{ij}(\tau) \cdot \big[1 - \inf(1, e_{a\to s_{ih}})\big] \right\}. \tag{14}$$

These two states represent the optimization of, respectively: the probability of target identification, ignoring the effects of sensor delay; and the actual probability of target identification. If one of the maxima does not exist, naturally only the remaining state — $x_{ip}(\tau)$ or $x_{id}(\tau)$ — is considered.

4) For each sensor, these optimal states are reduced in number from a maximum of $2W_1$ (since there are 1 or 2 from each of the W_1 time steps) to just one state per collection. This is done by forecasting the performance of each state with respect to its corresponding collection, ignoring all other targets. In this manner, an efficient search space $\{x_i'(\tau)\}$ is constructed for the first time step for each sensor.

5) All factorial combinations $s_q \in \mathbf{s}$ are constructed from the elements in the various search spaces $\{x_i'(\tau)\}$. Then, for each s_q, the system welfare in the first time step, $V_q(\tau)$, is calculated by Eq. (5). This begins the latter phase of the algorithm, in which the second window is used. Setting $W_2 = 2W_1$ is sufficient since for present purposes it is not necessary to observe beyond twice the maximum sensor delay: events which happen after this time can in every case be reacted to in sufficient time. In fact, more accurately,

$$W_2 = \left\lceil 2\sup_{i,b}\ e_{(a\to b)_i} \right\rceil. \tag{15}$$

6) To calculate $V_{q_{\text{total}}}(\tau) = V_q(\tau) + V_q(\tau+1) + \ldots + V_q(\tau+w) + \ldots + V_q(\tau+W_2-1)$, i.e. the sum of the system welfares over the second time window, the greedy algorithm is used, once per combination. (As an aside, note that $V_q(\tau+w)$ affects $V_q(\tau+w+1)$ since the fact that a target might be detected in one time step decreases the probability it will be detected in a subsequent time step.) The greedy algorithm runs as follows:
a) At the wth time step of the window, beginning with $w=1$, the possible state combinations are computed as per Step 3.
b) Two assumptions are made for the subsequent time steps: firstly, although $p_{ij}(\tau+w)$ will change and its changes will be tracked, the selected sensor-target allocations are temporarily assumed to remain the best choices throughout the window. Secondly, each sensor is able to observe the targets in its collection with negligible adjustments (i.e. little time is lost in changing states, once the change at the $(\tau+w)$th time step is made). These assumptions are what justify the use of a "greedy" algorithm, i.e. the algorithm is preparing to choose a *single* form of state change, as opposed to exploring all possible state changes as was done in the first time step.
c) A state combination is chosen to maximize the sum of the system welfares for all remaining time steps in the window, i.e. ${V_q}'(\tau+w) + {V_q}'(\tau+w+1) + \ldots + {V_q}'(\tau+W_2-1)$, where the primes ($'$) denote the fact that the system welfares are only temporarily approximated via the greedy algorithm. $V_q(\tau+w)$ is set to the corresponding value of ${V_q}'(\tau+w)$. The sensor movement is simulated.
d) Steps 6.a) through c) are repeated for the $(\tau+w+1)$th time step and so on, until the end of the time window ($w=W_2-1$).

The results of the above algorithm are exchanged using local messages as described in Section II. The candidate values of x_n in Eqs. (1–4) are given by $\{x_i'(\tau)\}$ in Step 4 for $i=n$, and the utilities $U(\mathbf{x}_m)$ in Eq. (4) are given by $V_{q_{\text{total}}}(\tau)$ for $s_q = \mathbf{x}_m$.

V. RESULTS

Messages converge typically after just one or two iterations. (In graph-colouring problems, where the optimization landscape is coarser, convergence often takes longer.)

In the first simulation, we reproduce the problem of Fig. 1. Sensors have fixed fields of view (120°) and slow rotation speeds. Sensor 3 cannot rotate in time to observe α. Hence, Sensor 1, which is already pointing in approximately the correct direction, adjusts its angle slightly in anticipation of the target (see Fig. 1b). Sensor 3 also adjusts its angle in preparation for observing α soon afterwards.

For the second simulation, real data was used as input. First, a lightly trafficked hectare was observed using a commercial off-the-shelf *Navtec* radar. Ranges and bearings of moving targets were collected at a sampling frequency of 1 Hz, which formed a database of 53 target trails.

TABLE I
SENSOR CHARACTERISTICS IN THE SECOND SIMULATION.

Sensing range	0–20 m	Max. targ. spd.	50 ms^{-1}
Rotation range	0–360°	Zoom range	20–90°
Rotation speed	$20°\mathrm{s}^{-1}$	Zoom speed	$60°\mathrm{s}^{-1}$

At nearly all time coordinates, a maximum of only one target trail was visible. However, a dynamic algorithm is useful primarily when multiple targets are visible simultaneously; hence, offline, the targets' time coordinates were artificially translated so that exactly five targets were visible at any time.

Each data point collected from the offline tracking algorithm consists of a 21-dimensional vector, whose components are a concatenation of the following:

- A (scalar) time stamp
- A two-dimensional position vector, (x, y)
- A two-dimensional velocity vector, (v_x, v_y)
- A 4×4 covariance matrix, S, for $\{x, y, v_x, v_y\}$

S is used to compute uncertainty in future target positions.

Six fictitious sensors were simulated *post hoc* with characteristics given in Table I. Where possible, these were taken from a mix of commercially available cameras, including the *480TVL* Pan/Tilt Dome Camera. The fictitious sensors were positioned at locations near the areas with highest traffic.

Once per second during the 178 seconds of target activity, the sensor-target allocations were decided using the DCF algorithm. The total number of successful target identifications during this period was recorded. This scenario was run 100 times and the median number of targets identified was 18 (range: 12–23).

The most important dynamic aspect of the algorithm is the forecasting of expected target identifications over a variable-length time window. This aspect was disabled, and the resulting simple non-dynamic algorithm was run another 100 times. The median number of targets identified was 14 (range: 7–19).

The results from the dynamic and non-dynamic algorithms are illustrated in Fig. 2. The notches on either side of the boxes indicate the uncertainties in the medians: the fact that the notches do not overlap vertically indicates that the medians of the two groups differ at the 5% significance level.

Finally, we used the same database to test the ability of the message-passing approach to find the globally optimal solution. At each of the 178 time steps, we computed the optimal sensor-target allocation using a centralized version of the DCF algorithm. Results from the message-passing algorithm precisely matched those of this centralized algorithm at every time step. (In addition, the computation time was found to be over 70% faster in the decentralized approach, at 68 rather than 232 ms/step/sensor.)

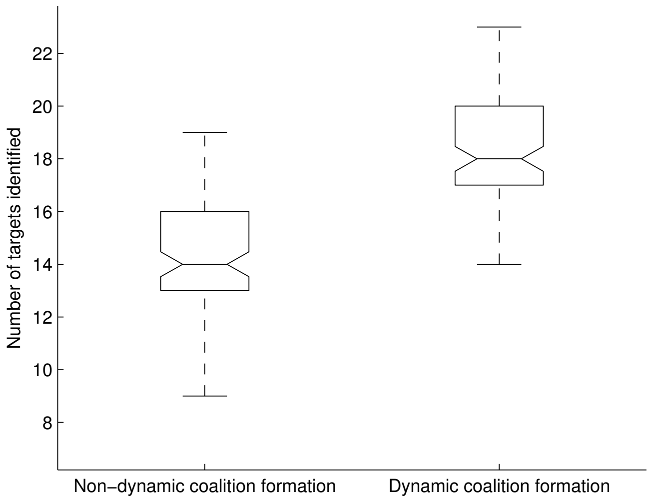

Fig. 2. A box-and-whiskers plot of the number of targets identified, summarizing 100 runs of the non-dynamic and dynamic coalition formation algorithms. Each box (hourglass shape) indicates the upper- and lower quartiles and the median; the whiskers (T-shaped) extend to the extremal data points, excluding outliers. (Outliers are data points separated from the box by a distance of more than 1.5 times the interquartile range.)

VI. DISCUSSION

Xiong and Svensson [16] reviewed the field of sensor management and framed their discussion, as some other authors have, within the JDL model of data fusion [14]. Specifically, sensor management comprises "Level 4" in this model. We have developed a Level-4 framework for the case of forecasts of arbitrary complexity being available to inform the allocation of sensors to targets. In a simulation based on real target movements, this approach leads to an improvement in the number of targets identified: approximately 18 rather than approximately 14. The extent of the improvement is significantly dependent on the scenario, but is guaranteed to be non-negative for the case of forecasts with correctly estimated uncertainties.

Our contribution is extensible to other sensor-network styles, such as cooperative unmanned vehicles, since the action space can be expanded arbitrarily. Elsewhere, we have explored the penalization of sensors for beginning and ending the observation of targets, rather than for changing their state; we have also developed the feature of identifying historically quiet or busy areas to better inform the algorithm and improve performance. Besides sensor networks, the framework is applicable to other multi-agent systems.

We have so far considered only two of the possibly five different time-varying parameters of interest when forming coalitions, according to Klusch and Gerber [6]. These five parameters are: the information available to sensors (e.g. the appearance of a new target, as in Fig. 1); the tasks to be accomplished (e.g. the value of observing targets can change); the computing resources; the number of sensors (e.g. some sensors may become damaged or be switched off); and the reliability of the sensors (e.g. as dictated by inclement weather). Projected changes in computing resources have not been explicitly taken into account in our algorithm since we assume that each sensor has dedicated computing power. The last two time-varying parameters are similar to one another if the absence of a sensor is treated as simply a sensor producing readings with zero confidence. Their impact might be modelled via a more detailed calculation of $v(C_j, t_j)$, to weight the sensor's contributions by a confidence measure; it is easy to adapt this to our configuration.

Future alternatives to the technology outlined in this paper might be expected to fall into the following categories of approach: search-based (of which ours is an example), information-theoretic, decision-theoretic, fuzzy-logic, or Markov-decision-problem approaches [16]. However, we have demonstrated that a search-based approach yields accurate solutions, and QinetiQ are currently investigating physical implementations of the algorithm.

ACKNOWLEDGMENTS

This work was undertaken within the ARGUS-II Defence and Aerospace Research Partnership, a collaboration among BAE SYSTEMS, QinetiQ, Rolls-Royce, and the Universities of Oxford and Southampton. ARGUS-II is funded by its industrial partners as well as the Engineering and Physical Sciences Research Council, the Ministry of Defence, and the Department for Business Enterprise and Regulatory Reform.

REFERENCES

[1] Y. Bar-Shalom and X.R. Li. *Multitarget-Multisensor Tracking: Principles and Techniques*. Artech House, 1995.
[2] V.D. Dang, R.K. Dash, A. Rogers, and N.R. Jennings. Overlapping coalition formation for efficient data fusion in multi-sensor networks. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, pages 635–40. AAAI-06, 2006.
[3] M. Ebden. Report B3: A simple DCF application. University of Oxford internal report, 2007.
[4] A. Farinelli, A. Rogers, A. Petcu, and N.R. Jennings. Decentralised coordination of low-power embedded devices using the max-sum algorithm. In *Proceedings of the Seventh International Conference on Autonomous Agents and Multi-Agent Systems*. AAMAS, 2008.
[5] C. Fernández, R. Béjar, B. Krishnamachari, and C. Gomes. Communication and computation in DisCSP algorithms. In *Eighth International Conference on Principles and Practice of Constraint Programming*, pages 1–15, 2002.
[6] M. Klusch and A. Gerber. Issues of dynamic coalition formation among rational agents. In *Second International Conference on Knowledge Systems for Coalition Operations*, pages 91–102. KSCO, 2002.
[7] H. Konishi and D. Ray. Coalition formation as a dynamic process. *J Econ Theory*, 110(1):1–41, 2001.
[8] C. Kreucher, K. Kastella, and A.O. Hero III. Sensor management using an active sensing approach. *Sig Proc*, 85:607–24, 2005.
[9] F.R. Kschischang, B.J. Frey, and H. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans Info Theory*, 47(2):498–519, 2001.
[10] D. Mackay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, fourth edition, 2003.
[11] J.M. Nash. Optimal allocation of tracking resources. In *Proceedings of the IEEE Conference on Decision and Control*, pages 1177–80, 1977.
[12] G.W. Ng and K.H. Ng. Sensor management — what, why and how. *Info Fusion*, 1(2):67–75, 2000.
[13] J. Schiff, D. Antonelli, A.G. Dimakis, D. Chu, and M.J. Wainwright. Robust message-passing for statistical inference in sensor networks. In *Proc. of the 6th International ACM/IEEE Symposium on Information Processing in Sensor Networks*, pages 109–18. IPSN, 2007.
[14] E. Waltz and J. Llinas. *Multisensor Data Fusion*. Artech House, 1990.
[15] M. Wooldridge. *An Introduction to MultiAgent Systems*. John Wiley and Sons, Ltd, 2002.
[16] N. Xiong and P. Svensson. Multi-sensor management for information fusion: issues and approaches. *Info Fusion*, 3(2):163–86, 2002.