Graph Marginalization for Rapid Assignment in Wide-Area Surveillance

Mark Ebden and Stephen Roberts

University of Oxford, Oxford, OX1 3PJ, United Kingdom
{mebden,sjrob}@robots.ox.ac.uk
http://www.robots.ox.ac.uk/~parg

Abstract. Decentralizing optimization problems across a network can reduce the time required to achieve a solution. We consider a wide-area surveillance (WAS) sensor network observing an environment by varying the state of each sensor so as to assign it to one or more moving objects. The aim is to maximize an arbitrary utility function related to object tracking or object identification, using graph marginalization in the form of belief propagation. The algorithm performs well in an example application with six heterogeneous sensors. In larger network simulations, the time savings owing to decentralization quickly exceed 90%, with no reduction in optimality.

Keywords: sensor networks, coalition formation, agents, max-sum algorithm, belief propagation.

1 Introduction

In wide-area surveillance (WAS), multiple sensor inputs are often integrated to assess moving objects for security purposes. Common sensor types are surface-movement radar, infrared cameras, closed-circuit television, and multispectral imaging. Up to about fifteen sensors might be distributed in the area we consider. Three aspects of interest about each object are its identity, position, and velocity, and in prioritizing the utility among these three, one may choose to:

- identify (i.e. assign to one of b bins) as many objects as possible,
- track the objects as accurately as possible (where accuracy may be measured as inversely proportional to a variance representing uncertainty, or otherwise),
- among the trespassers, identify as many as possible or track them as accurately as possible

In this paper, we have opted for the first of these for simplicity, without loss of generality. We simulate the identification of the objects as belonging to one of b categories (e.g. vehicle, person, etc.) without constraining ourselves to a specific algorithm of identification.

We approach the problem as one of dynamic coalition formation (DCF), a subject with great scope in economics and social sciences (see e.g. [2]). We have brought DCF to bear on sensor resource allocation, a subtopic of sensor management. Xiong and

J. Zheng et al. (Eds.): ADHOCNETS 2009, LNICST 28, pp. 691–703, 2010.

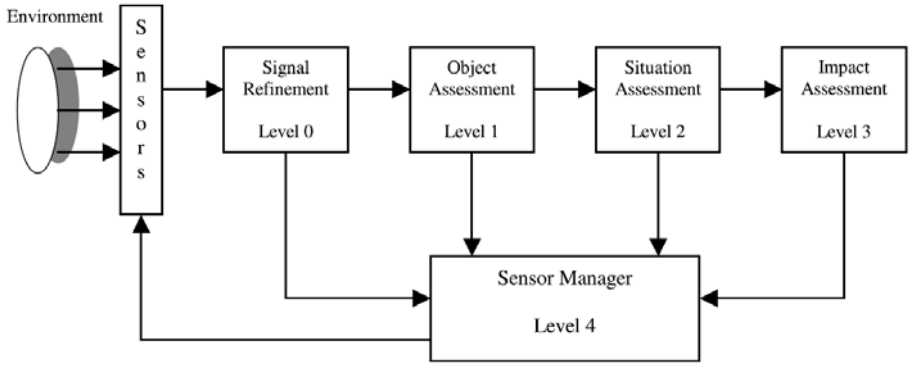

Fig. 1. The JDL model of data fusion from a sensor network. [9]

Svensson [9] reviewed the field of sensor management and framed their discussion, as some other authors have, within the JDL model of data fusion, which derives its name from the Joint Directors of Laboratories Data Fusion Group [8]. Specifically, sensor management comprises Level 4 in this model (see Figure 1); it can be instantiated in a centralized, decentralized, or hierarchical (hybrid) manner, and it can be pre-designed or it can incorporate machine learning techniques. Xiong and Svennson subdivide JDL Level 4 into five policy components (see Figure 2), and our work contributes to research in policy levels 1 and 2.

In particular, we have developed what is classified as a *search-based approach to sensor selection* [9], treating the allocation task as a search problem in some combinatorial space. Specifically, we are interested in the cases where sensors act as agents, i.e. they can choose among more than one action. For example, a camera has a limited instantaneous field of view (usually $<180°$), so its time-varying orientation and depth of view ought to be coordinated with those of its neighbours. We achieve this in a manner that scales well with the size of the network.

The next section describes how graph marginalization decentralizes the computation. Section 3 details the sensor network, and Section 4 gives the results of applying the algorithm to two types of scenario.

2 Decentralization of the Algorithm

Searches over combinatorial spaces can be prohibitively expensive without some form of decentralization, for example with belief networks. In our work, an agent is fed information about what other agents are doing, and chooses an action which helps attain the optimal global utility, in a very informed manner.

Our agents do not model explicitly every other agent. However, we achieve a commensurate overall effect with the max-sum algorithm, a form of message-passing algorithm in which information is shared locally among agents in an effort to solve a global problem [6]. Here, as stated previously, the global problem is to discover which states

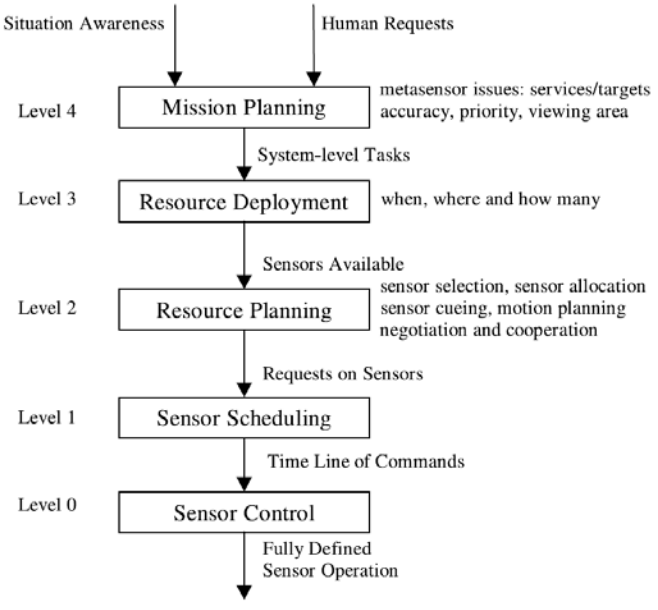

Fig. 2. Top-down problem solving by a sensor manager. [9]

the sensors in a sensor network should assume in order to maximize the expected number of objects identified in a given upcoming time window. 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 objects. It is based on factor graphs and an extension of the sum-product algorithm, also known as belief propagation. Farinelli *et al.* [5] pioneered the application of the max-sum algorithm in multi-agent coordination, and showed that the method works well in graph-colouring problems, implementing it in hardware using wireless, embedded devices by Chipcon. In principle the method can be extended 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 both observe at least one common object within some upcoming forecast window — 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 objects in the upcoming time window; non-integer estimates are also permitted. When several sensors point at the same object, 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.* [5]. Our method of selecting candidate

states x_n from a continuous space and of calculating the factors $U(\mathbf{x}_m)$ was described earlier [4].

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:

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

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

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

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:

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

It can be shown [5] that after convergence, the network's total utility is maximized approximately when each sensor assumes its optimal state x_n^* given by

$$x_n^* = \operatorname*{argmax}_{x_n} \left[\sum_{m\in\mathcal{M}(n)} R_{m\to n}(x_n) \right]. \tag{4}$$

The maximized utility is guaranteed to be optimal only when the sensors are connected as an "acyclic graph", e.g. as a tree. It should be noted that high levels of sensor noise, while decreasing the task performance, will not affect the convergence time or accuracy of the message-passing algorithm itself; only the values of the numbers being exchanged will change.

The use of local message passing is an attractive option primarily for problems that scale poorly with the number of sensors/agents. We test this structure on problems which can be calculated in a centralized or decentralized configuration.

3 The Sensor Network

Dang *et al.* [3] use a concise syntax to represent coalition formation (CF) problems, which we build upon. Differences between our sensor network and theirs include the following:

1. Our objects are moving, and the problem is one of DCF rather than CF. A description of the forecasting involved has been reported previously [4].
2. A less sophisticated pruning algorithm is performed, during exhaustive searches comparable to Algorithm #2 in [3].
3. No sleep states exist; i.e. the sensors are always observing.
4. The field of view (FOV) of each sensor is adjustable, to reflect for instance depth of view (zoom). This forces sensors to perform an accuracy tradeoff. Effecting a zoom-in or zoom-out also incurs a time penalty similar to those of the orientation angle changes.
5. Rather than letting each sensor assume one of a finite number of states, our possible states are continuous. We achieve this by assigning sensors to objects before, not after, exploring the utility of a sensor state.

In our work, the ith sensor, where $I = \{1, 2, ..., i, ..., n\}$, can assume any of an infinite number of states, reduced eventually to a finite subset customized to the environment, $s_i = \{0, 1, ..., k, ..., K_i - 1\}$, to track one or more of the m objects, $T = \{t_1, t_2, ..., t_j, ..., t_m\}$. All sensors $C \subseteq I$ tracking object t_j at a moment in time belong to the coalition (C, t_j), which has a utility $v(C, t_j)$ indicating how useful it is to track a particular object with a particular coalition. The set of all m coalitions existing at a particular time instant is the "coalition structure", CS $= \{(C_1, t_1), ..., (C_m, t_m)\}$.

3.1 Objects

On a two-dimensional grid, m heterogeneous moving objects are considered as objects, $T = \{t_1, t_2, \ldots, t_m\}$. Their locations are assumed to be known for N_τ discrete time steps, $\tau = \{1, 2, \ldots, N_\tau\}$. (These locations might be determined by surface radar, for instance.) The jth object is also assumed to move with a velocity function $u_j(\tau) = \big(u_{j_x}(\tau),\ u_{j_y}(\tau)\big)$ which represents a bounded two-dimensional Wiener process:

$$u_{j_x}(\tau) = \frac{u_{j_x}(\tau - 1) + \mathcal{N}(0, \sigma_u^2)}{\mu(\tau)} \qquad u_{j_y}(\tau) = \frac{u_{j_y}(\tau - 1) + \mathcal{N}(0, \sigma_u^2)}{\mu(\tau)} \tag{5}$$

$$\mu(\tau) = \sup\left\{1, \frac{|u_j(\tau)|}{u_{\max}}\right\}. \tag{6}$$

The normalization constant $\mu(\tau)$ ensures that the object's speed

$$|u_j(\tau)| = \sqrt{u_{j_x}(\tau)^2 + u_{j_y}(\tau)^2} \tag{7}$$

never exceeds an arbitrary value $u_{\max}$, and $\mathcal{N}(0, \sigma_u^2)$ represents a zero-mean normal distribution with a standard deviation of σ_u; in this paper $\sigma_u = u_{\max}/15$.

These assumptions are arguably too strong for many real data sets, and could be replaced by a variable structure interacting multiple model (VS-IMM), jump Markov linear system (JMLS), or similar. However, they suffice for present purposes, and allow simple forecasts of the position vector:

$$\big(x_{t_j}(\tau),\ y_{t_j}(\tau)\big) = \big(x_{t_j}(\tau - 1) + u_{j_x}(\tau),\ \ y_{t_j}(\tau - 1) + u_{j_y}(\tau)\big). \tag{8}$$

3.2 Sensors

Although the objects' locations and velocities are known, their identities are not. Thus, n heterogeneous sensors, $I = \{1, 2, \ldots, n\}$, are placed in the grid to observe the objects and gradually determine their identities. If the distance between the ith sensor, positioned at (x_i, y_i), and the jth object is

$$d_{ij}(\tau) = \sqrt{\left[x_i - x_{t_j}(\tau)\right]^2 + \left[y_i - y_{t_j}(\tau)\right]^2}, \tag{9}$$

and the field of view (e.g. reflecting degree of zoom) is z_i, measured in radians, then the chance that the object may be identified within any time step is some function of these three parameters. Generally, the likelihood of a successful identification decreases with the speed of an object, the sensor-object distance, and the field of view of the sensor — see Figure 3 for an example graph. For identification to occur, the object must also be visible to the sensor; this will be described in Section 3.4. The effects of sensor noise can be easily incorporated by lowering the value of $p_{\max}$ in Figure 3.

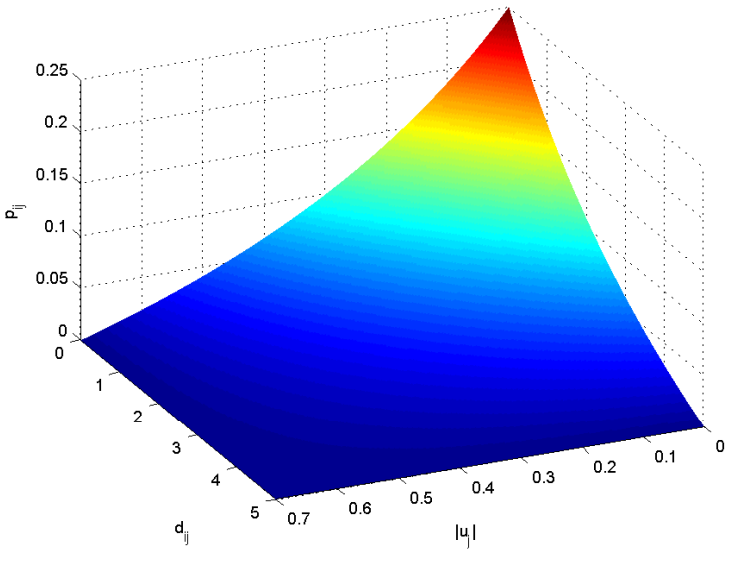

Fig. 3. An example of the probability distribution p_{ij} that the ith sensor will identify the jth object, as a function of the object speed $|u_j|$ and the separation distance d_{ij}. Here, $v_{\max} = 0.7$, $d_{\max} = 5$, and $p_{\max} = 0.25$.

The probability of identifying the jth object considers simultaneously all sensors in its coalition. This allows for more sophisticated object identification probability distributions, if these are ever desired. As a simple example, the minimum number of sensors in a coalition can be specified. The sensors may be of different types.

3.3 Objective

The aim is to identify as many objects as possible within the N_τ time steps. The jth object is observed by a coalition $C_j(\tau)$ of $n_{t_j} \in \mathbb{W}$ independent sensors, which has a time-dependent utility given by the probability that the object will be identified. In our work this utility is simplistically defined as

$$v\big(C_j(\tau)\big) = 1 - \prod_{i \in C_j(\tau)} \left[1 - p_{ij}(\tau)\right], \tag{10}$$

where p_{ij} is calculated independently for each sensor-object pair; however, clearly this can be extended to reflect sensor mutualism more directly. The total utility of all coalitions equals the expected number of objects which will be identified in a particular time step, before considering the effects of sensor movement or of new objects appearing. This total utility is affected significantly by the number of coalitions each sensor is permitted to belong to, $m_{\text{sensor}} \in \mathbb{W}$. For this paper $m_{\text{sensor}} = m$ without loss of generality.

Positions and velocities are revealed to all sensors sequentially, from which future values can be extrapolated. To account for the uncertainty, $p_{ij}(\tau)$ is calculated as:

$$p_{ij}(\tau) = \sum_{u_j(\tau)} \sum_{d_{ij}(\tau)} \sum_{f_{ij}(\tau)} \left[p_{ij}(\tau) | u_j(\tau), d_{ij}(\tau), f_{ij}(\tau)\right] \, p\big(u_j(\tau)\big) p\big(d_{ij}(\tau)\big) p\big(f_{ij}(\tau)\big) \tag{11}$$

where

$$f_{ij}(\tau) = \begin{cases} 1 & \text{if the } j\text{th object is within the field of view of the } i\text{th sensor,} \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

At present, the model does not marginalize over $u_j(\tau)$ or $d_{ij}(\tau)$, but it does marginalize over the indicator variable; see next subsection.

3.4 Probability That an Object Will Be Visible to a Sensor

The probability that the jth object falls within the circle sector monitored by the ith sensor is $p\left(f_{ij}(\tau)\right)$ in (11). Recall that object velocities are modelled as two-dimensional modified Wiener processes. Hence, given an object's position (x_0, y_0) and previous velocity (u_{x0}, u_{y_0}), the estimate of its position (x_1, y_1) at the next time step can be approximated as

$$x_1 = \inf\left\{x_0 + \upsilon_{\max}\Delta\tau,\ \sup\left\{\mathcal{N}(x_0 + u_{x0}\Delta\tau, \sigma_u),\ x_0 - \upsilon_{\max}\Delta\tau\right\}\right\}$$
$$y_1 = \inf\left\{y_0 + \upsilon_{\max}\Delta\tau,\ \sup\left\{\mathcal{N}(y_0 + u_{y_0}\Delta\tau, \sigma_u),\ y_0 - \upsilon_{\max}\Delta\tau\right\}\right\}$$

provided that $\upsilon_{\max}\Delta\tau \gtrsim 3\sigma_u$. Thus if $(\mu_x, \mu_y) = (x_0 + u_{x0}\Delta\tau, y_0 + u_{y_0}\Delta\tau)$ lies more than $\upsilon_{\max}\Delta\tau$ units away from the boundary of the circle sector, $p\left(f_{ij}(\tau)\right)$ can be quickly determined. In considering the remaining cases, the sector boundary is decomposed into an arc and two line segments. When the point is close to the arc, the exact calculation of $p\left(f_{ij}(\tau)\right)$ is unnecessary since $p_{ij}(\tau)$ is negligible in that region; hence

$p\left(f_{ij}(\tau)\right)$ is approximated as 0 outside the sector and 1 inside. More interesting is when the point lies close to either of the two line segments.

To solve this final case, it is helpful to note that the perpendicular distance from a point (μ_x, μ_y) to a line defined by $ax + by + c = 0$ is given by

$$d = \frac{|a\mu_x + b\mu_y + c|}{\sqrt{a^2 + b^2}}. \tag{13}$$

If the line has slope m and intersects an arbitrary point (x_s, y_s), then the point-slope form yields $a = -m$, $b = 1$, and $c = mx_s - y_s$. Letting the line represent one of the edges of an F-radian-wide field of view for a sensor oriented to angle θ, then

$$m = \tan\left(\theta \pm \frac{F}{2}\right). \tag{14}$$

Hence if a sensor in the kth state is situated at (x_s, y_s), then the distance between an object at (μ_x, μ_y) and either of the two edges of the field of view is given by

$$\begin{aligned} d &= \frac{|-m\mu_x + \mu_y + mx_s - y_s|}{\sqrt{m^2 + 1}} \\ &= \frac{|-\tan\left(\theta \pm \frac{F}{2}\right)\mu_x + \mu_y + mx_s - y_s|}{\sqrt{\tan^2\left(\theta \pm \frac{F}{2}\right) + 1}} \\ &= \frac{|-\tan\left(\theta \pm \frac{F}{2}\right)\mu_x + \mu_y + mx_s - y_s|}{\sec\left(\theta \pm \frac{F}{2}\right)}. \end{aligned}$$

Finally, if $\Phi(\cdot)$ is the standard normal cumulative distribution function, and d_1 and d_2 are the distances from the object to each of the two line segments, then a consideration of the marginals of a bivariate Gaussian leads to

$$p\left(f_{ij}(\tau)\right) = \begin{cases} \Phi\left(\frac{d_1}{\sigma_u}\right) + \Phi\left(\frac{d_2}{\sigma_u}\right) - 1 & \text{if the object is inside the sector,} \\ 1 - \Phi\left(\frac{\inf(d_1, d_2)}{\sigma_u}\right) & \text{if the object is outside the sector.} \end{cases} \tag{15}$$

One final case needs to be considered: that in which the object is not within the sensor's field of view but $d_1 < v_{\max}\Delta\tau$ and $d_2 < v_{\max}\Delta\tau$. In this case, a Monte Carlo simulation of 100 runs is used to estimate $p\left(f_{ij}(\tau)\right)$.

4 Simulations

4.1 Data Structure

Each data point collected for our simulations consists of a 21-dimensional vector, whose elements can be reassembled easily to recover the following: a 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 $\{x, y, v_x, v_y\}$. Thus S presents the information needed to form

a picture of the current environment and to compute the uncertainty in future object positions. Let (z_x, z_y) represent the object position one time step in the future. A simple way to estimate this position is $(z_x, z_y) = (x + 1 \cdot v_x, y + 1 \cdot v_y)$. This can be represented by a bivariate Gaussian distribution centred on $(\overline{z_x}, \overline{z_y}) = (\overline{x} + \overline{v_x}, \overline{y} + \overline{v_y})$ with a covariance matrix given by

$$\Sigma = \begin{bmatrix} \mathrm{var}(x) + \mathrm{var}(v_x) + 2\mathrm{cov}(x, v_x) & -\mathrm{cov}(x, v_x) - \mathrm{cov}(y, v_y) \\ -\mathrm{cov}(x, v_x) - \mathrm{cov}(y, v_y) & \mathrm{var}(y) + \mathrm{var}(v_y) + 2\mathrm{cov}(y, v_y) \end{bmatrix}. \quad (16)$$

The information in Σ is currently used in a conservative way. The larger of the two diagonal elements is taken as the variance for a circular, not elliptical, Gaussian around $(\overline{z_x}, \overline{z_y})$. This approach simplifies the geometrical problems which arise in assessing marginal probabilities of identification, but can be refined as needed.

4.2 First Application

The algorithm has been found to run well in several scenarios. The first example is an offline analysis of real trajectories of people and road vehicles in an industrial setting.

A commercial off-the-shelf Navtec radar, with a sampling frequency of 1 Hz, was used to observe an area of approximately one hectare. Data were collected over the course of an hour; however, to increase the challenge, the objects' time coordinates were artificially translated so that, in subsequent offline analysis, exactly five objects were visible at any time (see Figure 4). The five always comprised some evolving mixture of

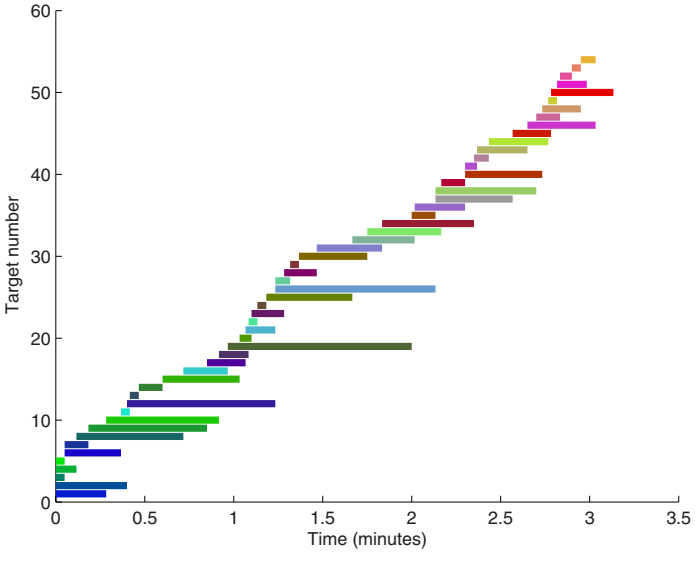

Fig. 4. Adjusted object occurrence times. The time axes of objects have been shifted to increase the incidence of object simultaneity, which was initially rare.

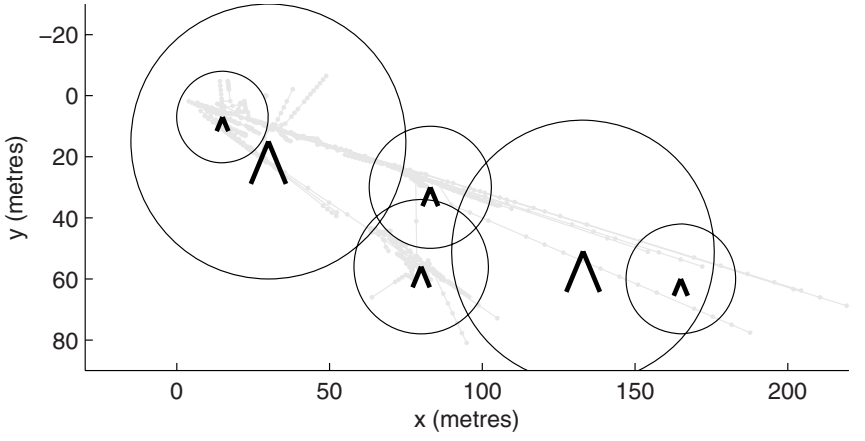

Fig. 5. Placement of six simulated sensors (circled chevrons) to observe objects moving along trajectories (faint lines)

cars, trucks, and pedestrians. In total, approximately 50 target trails were used, over a compressed period of nearly three minutes.

In the offline analysis, simulated sensors were positioned at locations near areas of high traffic, as illustrated in Figure 5. The circles in this figure indicate the range of each sensor, which varied from 15–45 metres. Although the two sensors in the upper left portion of the figure might seem at first to be adequately governable in a separate network, they formed neighbourhoods with the other sensors and communicated with them. This was due to the dynamic nature of the algorithm: one sensor's planned future movements would influence the decision making of sensors slightly out of range, since in an upcoming time window, a given target could be observed by both of them, albeit not simultaneously.

Each sensor was able to rotate 360°, at a rate of 100° per second. Sensors could zoom in and out to varying degrees: in each case the minimum field of view was between 5° and 30°, and the maximum field of view was between 45° and 180°. The speed at which this angle changed was 60° per second for all sensors. The sensors were thus homogeneous in certain characteristics and heterogeneous in others, although it would have been straightforward to make them entirely heterogeneous in all characteristics, including their ranges, sensing capabilities, and reaction times.

The decentralized algorithm was run sequentially at each time step, with information about which targets were successfully identified stored in memory. The total number of targets identified was also gradually tallied. The experiment was repeated ten times, and the number of targets identified ranged from 32 to 39 with a median of 33. The computation time averaged 134 ms/step/sensor.

Results from the decentralized algorithm were consistently optimal, as demonstrated by comparison with a centralized exhaustive-search algorithm. The algorithm typically converged after just one or two message-passing iterations. In graph-colouring problems, where the optimization landscape is coarser, we have found that convergence often takes several steps.

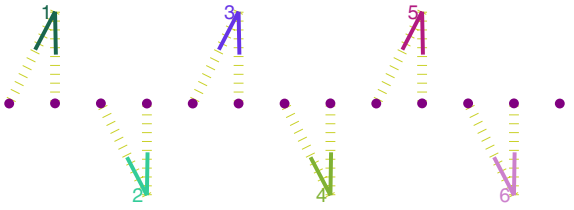

Fig. 6. An example of the artificial sensor network used to characterize the scaling properties of the algorithm. The case for $n = 6$ links in the chain is shown, where numbers indicate the six sensors and circles indicate the thirteen objects. Dashed lines indicate the sensor-object assignments in the optimal solution.

4.3 Second Application

The algorithm was then run on a more rigidly structured, artificial sensor network (see Figure 6), to test its scaling properties.

Each "link" in the pictured chain consists of two nearly stationary objects and one sensor. In addition, an object was appended to one end of the chain so that each sensor would have the same state-space size; hence for n links in the chain there are n sensors observing $2n + 1$ objects. (The additional object was unnecessary since sensors are obviously allowed to consider unequal numbers of options; the object was added only in order to more smoothly examine computational scaling as the chain grew. To ensure that it did not increase the number of optimal solutions beyond unity, the object was displaced slightly.)

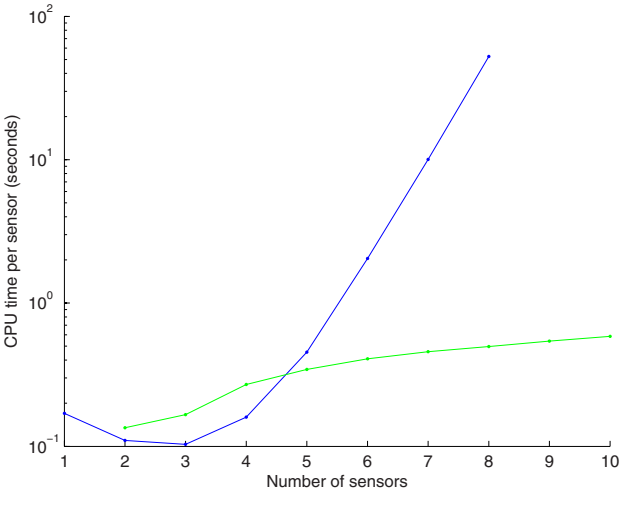

Fig. 7. An illustration of the scaling properties of the decentralized algorithm (lower line) versus the centralized algorithm (upper line), for the artificial sensor network described in Figure 6

It was found that as the number of sensors in the network was incremented, the decentralized algorithm was able in every case to find the optimal state as computed by a centralized exhaustive-search algorithm. (In further exploration on other artificial networks, we have still not witnessed a failure of the decentralized algorithm to find the optimal solution.) In addition, as is evident from Figure 7, the computational effort did not increase nearly as rapidly as that for the centralized algorithm. The initial decrease in the computation time for the centralized algorithm reflects the fact that certain overhead is associated for setting up the problem, and is a less interesting feature of the graph than the characteristics observed for three or more sensors. Moreover, for seven or more sensors, the time savings always exceeded 90%. The CPU times were recorded from runs on a normal desktop machine, the Intel® Core™2 CPU 6700 (2.66 GHz).

5 Conclusion

The decentralized approach presented in this paper yields substantial reductions in computational complexity. In optimization problems in general, the decentralization of computations across a network can greatly reduce the time required to achieve a solution. In our current application of WAS, in which an environment is observed by varying the state of each sensor, the aim was to maximize a utility function related to object identification. We have shown that in a network with six sensors, the algorithm works well, and in a second application in which the network was allowed to grow without limit, the computational savings from decentralization quickly exceeds 90%.

These savings contribute to "power-aware computation", an important issue in wireless sensor network power control [7]. In terms of the sensor networks protocol stack [1], the savings can be thought of as being achieved via improvements in the Application Layer and the Task Management Plane.

Some extensions to the WAS environment which could be made include the following:

- Allowing the probability of target identification to rely on previous observations
- Marginalizing over a joint probability distribution of $d_{ij}(\tau_{\text{future}})$ and $u_j(\tau_{\text{future}})$
- Extending the tracking capabilities and allowing the present object locations and velocities to not be known precisely (the sensors might be equipped with Kalman filters to maintain estimates of these quantities)
- Replacing the constant-velocity model with a variable structure interacting multiple model (VS-IMM), jump Markov linear system (JMLS), or similar.

Meanwhile, we are pursuing the implementation of extended versions of this algorithm in networks of video cameras with adjustable viewing states.

Acknowledgments

This research was undertaken as part of the ARGUS II DARP and ALADDIN projects. ARGUS II DARP (Defence and Aerospace Research Partnership) is a collaborative project involving BAE SYSTEMS, QinetiQ, Rolls-Royce, Oxford University and Southampton University, and is funded by the industrial partners together with the Engineering and Physical Science Research Council (EPSRC), Ministry of Defence, and the

Department for Business Enterprise and Regulatory Reform. ALADDIN (Autonomous Learning Agents for Decentralised Data and Information Systems) is jointly funded by a BAE Systems and EPSRC strategic partnership (EP/C548051/1).

References

1. Akyildiz, I., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. IEEE Communications Magazine 40(8), 102–114 (2002)
2. Breban, S., Vassileva, J.: Long-term coalitions for the electronic marketplace. In: Proceedings of the E-Commerce Applications Workshop, Canadian AI Conference, June 7-10, pp. 1–7 (2001)
3. Dang, V.D., Dash, R.K., Rogers, A., Jennings, N.R.: Overlapping coalition formation for efficient data fusion in multi-sensor networks. In: Proceedings of the Twenty-First National Conference on Artificial Intelligence, AAAI 2006, pp. 635–640 (2006)
4. Ebden, M., Briers, M., Roberts, S.: Decentralized predictive sensor allocation. In: Proceedings of the IEEE Conference on Decision and Control, pp. 1702–1707 (2008)
5. Farinelli, A., Rogers, A., Petcu, A., Jennings, N.R.: 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)
6. Mackay, D.: Information Theory, Inference, and Learning Algorithms, 4th edn. Cambridge University Press, Cambridge (2003)
7. Pantazis, N., Vergados, D.D.: A survey on power control issues in wireless sensor networks. IEEE Communications Surveys and Tutorials 9(4), 86–107 (2007)
8. Waltz, E., Llinas, J.: Multisensor Data Fusion. Artech House (1990)
9. Xiong, N., Svensson, P.: Multi-sensor management for information fusion: issues and approaches. Info Fusion 3(2), 163–186 (2002)