Overlapping Coalition Formation for Efficient Data Fusion in Multi-Sensor Networks

Viet Dung Dang, Rajdeep K. Dash, Alex Rogers and Nicholas R. Jennings
School of Electronics and Computer Science,University of Southampton
Southampton, SO17 1BJ UK
Email:{vdd,rkd02r,acr,nrj}@ecs.soton.ac.uk

Abstract

This paper develops new algorithms for coalition formation within multi-sensor networks tasked with performing wide-area surveillance. Specifically, we cast this application as an instance of coalition formation, with overlapping coalitions. We show that within this application area sub-additive coalition valuations are typical, and we thus use this structural property of the problem to derive two novel algorithms (an approximate greedy one that operates in polynomial time and has a calculated bound to the optimum, and an optimal branch-and-bound one) to find the optimal coalition structure in this instance. We empirically evaluate the performance of these algorithms within a generic model of a multi-sensor network performing wide area surveillance. These results show that the polynomial algorithm typically generated solutions much closer to the optimal than the theoretical bound, and prove the effectiveness of our pruning procedure.

Introduction

Coalition formation (CF) is the coming together of a number of distinct, autonomous agents in order to increase their individual gains by collaborating. This is an important form of interaction in multi-agent systems because many applications require independent agents to come together for a short while to solve a specific task and disband once it is complete. As such, it has recently been advocated for task allocation scenarios where groups of agents derive a certain value (and/or cost) from tasks being performed in the coalition (Shehory & Kraus 1998). Building on this, in this paper, we apply CF to one such scenario, namely wide-area surveillance by autonomous sensor networks (e.g. performing monitoring and intruder detection in areas of high security). This is an important application that has received renewed interest in recent years, and a key question within this field, is how to coordinate multiple sensors in order to focus their attention onto areas of interest, whilst balancing the need for both coverage and precision. Thus the problem can naturally be modelled as one of CF since a number of groups of sensors need to be formed to focus on particular targets of interest, these groupings combine their resources for the group's benefit and then they disband when the target is no longer present or a more important one appears.

However in order to apply CF within this domain, we need to extend the current state of the art. Specifically, to date, much of the research within this area has assumed non-overlapping coalitions in which agents are members of at most one coalition (see section Related Work for more details). Now, in the multi-sensor networks that we consider, this assumption no longer holds. Since, the sensors can track multiple targets simultaneously, multiple overlapping coalitions can be formed. Thus, against this background, this paper advances the state of the art in the following ways:

- We cast the problem of sensor coordination for wide-area surveillance as a coalition formation process, and show that, in general, this results in a coalition formation problem in which multiple coalitions may overlap and in which the coalition's values are typically sub-additive.
- We develop two novel algorithms to calculate the optimal coalition structure when faced with overlapping coalitions and sub-additive coalitional values. The first is a polynomial time approximate algorithm that uses a greedy technique and has a calculated bound from the optimum (Cormen, Leiserson, & Rivest 1990). The second is an optimal algorithm based on a branch-and-bound technique (Land & Doig 1960). We evaluate the performance of these algorithms in a generic setting, and show that the typical performance of the polynomial algorithm is typically much better then the calculated bound. In addition, we show that the optimal branch-and-bound algorithm is able to effectively prune the search space.

The rest of the paper is organised as follows. In the next section we describe the wide-area sensing scenario that motivates this work. Following this, we present our two algorithms for finding the optimal coalition structure in our overlapping coalition scenario. We empirically assess the performance of these algorithms in the following section, and finally, we conclude and discuss future work.

The Coalition Model

We now present our model of coalitions within a sensor network applied to wide-area surveillance. Thus, our model consists of a set of n sensors, $I = \{1, 2, ...n\}$, and a set of m targets, $T = \{t_1, t_2, ...t_m\}$, within an area environment that the sensors are tasked with monitoring. Each sensor i has K_i possible states, and $s_i \in \{0, \ldots, k, \ldots K_i - 1\}$ denotes

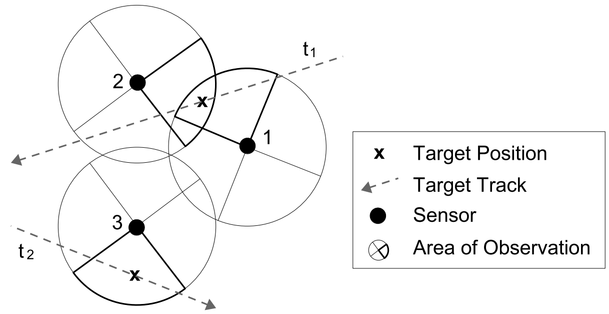

Figure 1: An example sensor network in which three sensors (1,2 and 3) track the position of two targets (t_1 and t_2) that pass through their field of view (indicated in bold).

the state it is in. When $s_i = 0$, the sensor i is the 'sleep' state (i.e. a state in which it is not sensing). The remaining $K_i - 1$ states are sensing states that indicate the directional capability of the sensor (i.e. an individual sensor can orientate its focus of attention into a number of distinct regions). Thus, depending on the sensing state that each sensor adopts, the sensor network as a whole may focus the attention of different combinations of sensors on to different targets. Figure 1 shows a simple instantiation of such a sensor network. Here there are three sensors, $I = \{1, 2, 3\}$ that are tracking two targets, $T = \{t_1, t_2\}$, within their field of view. The sensors can either sleep or orientate their sensing in one of four directions. Since they have a fixed range, they can thus focus their attention within one of four sectors centered on the sensor itself (the active sector is shown in bold in the diagram).

Now, a coalition in this scenario, is a group of sensors tracking a particular target (e.g. sensors 1 and 2 tracking target t_1 in the example shown in figure 1). Let $visibility(i, s_i, t_j)$ be a binary logical variable such that it is $true$ if target t_j can be observed by sensor i when in state s_i, and $false$ otherwise. Then we can define a coalition as:

Definition 1 ***Coalition.*** *A coalition is a tuple* (C, t_j) *whereby* $C \subseteq I$ *is a group of sensors such that* $\forall i \in C$, *sensor* i *is in state* s_i *such that* $visibility(i, s_i, t_j) = true$.

Note that from the above definition, when an agent chooses to be in a particular state, it becomes a member of those coalitions that are responsible for tracking all the targets that are visible in that state, and thus, the sensor may be a member of several overlapping coalitions (this would occur in our example if another target fell within the active sensing sectors of both sensor 1 and 2).

Definition 2 ***Overlapping Coalitions.*** *Two coalitions* (C, t_j) *and* (D, t_l) *are overlapping if* $C \cap D \neq \emptyset$

Now, each coalition (C, t_j) has a value $v(C, t_j)$ that represents the value of having a number of sensors tracking a target (we discuss in the next section how this value is calculated). In addition, each sensor incurs costs depending on the sensing state that it has adopted. For example, the cost maybe zero when the sensor is turned on and non-zero otherwise. Moreover, in this paper, we are interested in the system welfare as it is an effective indication of the system's performance, especially in cooperative environments. Thus, the optimal coalition structure generation problem is to find a set of coalitions $CS^* = \{(C_1, t_1), \ldots, (C_m, t_m)\}$ such that the system welfare is maximised:

$$CS^* = \arg \max_{CS \in \Gamma(I,T)} \left[\sum_{t_j \in T} v(C_j, t_j) - \sum_{i \in I} c_i \right] \quad (1)$$

where $\Gamma(I, T)$ is the set of all possible coalition structures given the targets and the agents. Note that unlike the standard coalition model, in our formalism we can not simply incorporate the costs into the coalition values. Doing so would incur multiple counting of the costs, since whilst there may be m coalitions representing each target, there are n sensors (and these sensors incur costs depending on their sensing state rather than the number of coalitions of which they are members).

Coalition Values

Now, since wide-area sensing is concerned with information gathering, it is natural to consider a coalition valuation function based on the information content of observations. In this case, the goal of the sensor network when coordinating the focus of individual sensors, is to obtain the maximum information from the environment. A common way to measure information in target tracking scenarios is to use Fisher information; a measure of the uncertainty of the estimated position of each target (Dash, Rogers, Reece, Roberts and Jennings 2005). Such a measure is attractive because when a number of sensors observe the same target and then fuse their individual estimates, the information content of the fused estimate is simply given by the sum of the information content of the individual un-fused estimates. Thus, when the coalition value is represented as the information content of position estimates, the coalition values are additive. For example, in figure 1 where both sensors 1 and 2 observe and fuse information about target 1, then $v(\{1, 2\}, t_1) = v(\{1\}, t_1) + v(\{2\}, t_1)$.

However, this additivity only applies when the individual estimates are independent. A more likely scenario within sensor networks is that these individual estimates are correlated to some degree. This will typically occur either through the exchange and fusion of earlier position estimates, or alternatively, by sensors using shared assumptions (such as a common model of the target's motion). Now, when these estimates are correlated, the coalition values become sub-additive (Reece and Roberts 2005). That is, due to the correlation, the fused estimate contains less information than the sum of the individual estimates, and thus, in our example, $v(\{1, 2\}, t_1) < v(\{1\}, t_1) + v(\{2\}, t_1)$.

The same sub-additive valuation also occurs in other more general models of sensor networks. For example, in (Lesser, Ortiz & Tambe 2003), the authors explicitly impose such sub-additivity when the number of sensors observing a single target increases. Indeed, within our overlapping coalition setting, such sub-additive coalition values are very common, and occur whenever there are diminishing returns as more resources are applied to a task.

Thus, in this paper, we focus on coalition values that obey the following two conditions:

- Monotonicity: $v(C, t_j) \leq v(D, t_j)$ if $C \subseteq D$
  This ensures that adding new members to a coalition can never reduce its value. In our case, this implies that obtaining observations from more sensors about a target cannot decrease the coalition value.
- Sub-additivity: $v(C \cup D, t_j) \leq v(C, t_j) + v(D, t_j)$
  This implies that if two different groups of sensors track a certain target t_j, then the sum of the value each derives is never less than if the union of the two groups perform it. In our scenario, suppose a group of sensors C tracks a target t_j. Now suppose another group of sensors D track the same target. Then the sum of the value of these two exclusive events cannot be less than if all the members of C and D joined to track the same target. Sub-additivity intuitively occurs when $C \cap D \neq \emptyset$ or due to the diminishing returns each new member brings to a coalition.

Sensor Costs

As described above, the costs of the sensors are calculated separately from the coalition value. In the case of simple sensors that incur a fixed cost dependent on their state s_i, we can model the cost as:

$$c_i = \begin{cases} 0 \text{ if } s_i = 0 \\ cost_i \text{ otherwise} \end{cases}$$

In more complex settings, the sensor cost may also reflect the additional costs incurred when changing from one sensing state to another (e.g. the cost of changing its orientation to track another set of targets), or reflect the fact that in battery power devices the cost of sensing may depend on the state of charge of the battery. However, in this paper, we consider the simple cost structure since this issue has no impact on the performance of our coalition formation algorithms, which we now describe.

Coalition Formation Algorithms

In this section, we present our two coalition formation algorithms. Specifically, we describe a fast polynomial, approximate algorithm (that can produce a solution within a finite bound of the optimal), and then an optimal branch-and-bound algorithm.

The Polynomial Algorithm

Algorithm 1 is a polynomial time algorithm that produces an approximation of the optimal solution (see figure 2). Basically, it operates in a greedy manner. It first chooses the best action by a sensor (e.g. the action that brings the biggest value) (see step 1). In the second step, it chooses the best action of another sensor taking the first sensor into consideration. Then, in the third step, it chooses the best action of another sensor taking the first two sensors into consideration. The process then repeats until there is no sensor left. We can now analyse the algorithm to assess its properties.

Theorem 1.1 *The complexity of algorithm 1 is $O(n^2 m)$.*

Algorithm 1

1. *Each agent chooses its best state taking only itself into consideration and calculates this best outcome. That is, each agent i chooses its state s_i such that:*

$$\sum_{\substack{j=1 \\ visibility(i,s_i,t_j)=true}}^{m} v(\{i\}, t_j) - c_i \text{ is maximised}$$

and calculates this best personal outcome (denoted p_i). The agents then choose agent i_1 with the best outcome:

$$p_{i_1} = \max_{i \in I} p_i$$

Agent i_1 switches to its best state if not in that state yet.

2. *Each agent, except i_1, chooses its best state taking only itself and i_1 into consideration and calculates this best outcome. That is, each agent i chooses its best state s_i such that:*

$$\sum_{\substack{j=1 \\ C_j \subseteq \{i_1, i\}}}^{m} v(C_j, t_j) - c_{i_1} - c_i \text{ is maximised}$$

and calculate its best outcome p'_i. The agents then choose agent i_2 with the best outcome:

$$p'_{i_2} = \max_{i \in I, i \neq i_1} p'_i$$

Agent i_2 switches to its best state if not in that state yet.

3. *Repeat the above step until we reach the last agent.*

Figure 2: The polynomial coalition formation algorithm.

PROOF. At each step, it requires to get through $O(n)$ sensors to find the best action of a sensor. For each sensor, we have to calculate the outcome for each state by summing $O(m)$ coalition values together. As there are n steps, the complexity is $O(n^2 m)$. □

Theorem 1.2 *The solution of algorithm 1 is within a bound n of the optimal. That is, given that V_1 is the system welfare of the solution of algorithm 1 and V^* the optimal solution:*

$$\frac{V^*}{V_1} \leq n$$

PROOF. Let $\langle s_i^* \rangle_{i=1}^n$ be the optimal *state vector* (that is, the vector contains the states of all sensor agents). For $1 \leq j \leq m$, let C_j^* be the coalition of sensors that track target t_j associated with the optimal solution. The optimal solution's system welfare V^* then is:

$$\begin{aligned} V^* &= \sum_{j=1}^{m} v(C_j^*, t_j) - \sum_{i=1}^{n} c_i \\ &\leq \sum_{j=1}^{m} \sum_{i \in C_j^*} v(\{i\}, t_j) - \sum_{i=1}^{n} c_i \end{aligned}$$

$$\begin{aligned}\Rightarrow V^* &\leq \sum_{i=1}^{n}\sum_{j=1}^{m} v(\{i\}, t_j) - \sum_{i=1}^{n} c_i \\ &\leq \sum_{i=1}^{n}(\sum_{j=1}^{m} v(\{i\}, t_j) - c_i) \\ &\leq n * (\sum_{j=1}^{m} v(\{i_1\}, t_j) - c_{i_1}) \\ &\leq n * V_1\end{aligned}$$

□

The Optimal Algorithm

The optimal algorithm is a branch-and-bound algorithm that finds the optimal solution. First, however, we define the concept of a *weak state* as it will be used in the algorithm.

Definition 3 *A state s_i of agent i is called a weak state iff agent i in state s_i does not see any target in its range.*

Proposition 1 *A state vector $\langle s_1, s_2, ..., s_n\rangle$ is not optimal if there exists an i such that s_i is a weak state.*

PROOF. This is trivial due to the fact that $V(s_1, s_2, ..., s_n\rangle$ is always less than $V\langle s_1, ..., s_{i-1}, 0, s_{i+1}, ..., s_n\rangle$ (0 means the sleeping state). □

We present the branch-and-bound algorithm in figure 4. This basically searches through the search space in a depth-first search manner, then uses a branch-and-bound technique to prune a subtree whenever possible. Specifically, if we reach a node and the upper bound value of all nodes that branch under that node is less than or equals the current best solution, we can prune the whole subtree under the node. The original best solution is the solution of the greedy algorithm, while the upper bound of the subtree is derived from the sub-additivity as detailed in figure 4. Also see figure 3 for an example search tree in case $n = 3$ and $k_i = 2$, for all $1 \leq i \leq n$.

Now one of the main issues that affects the performance of Branch-and-Bound algorithms is choosing the tree structure. To this end, we present a process for selecting the tree structure (which in this case is equivalent to an ordering of the agents) with which the algorithm will likely prune the subtrees quickly (see figure 5). Basically, it contains 2 phases. In the first one, all agents with weak states are chosen first and ordered decreasingly according to the number of their weak states. The idea is to maximise the number of pruned subtrees early on (due to proposition 1). In the second phase, the remaining agents are ordered in a similar way to the greedy algorithm (but in reverse). That is, it first chooses the worst action by a sensor. In the second step, it chooses the worst action of another sensor taking the first sensor into consideration. Then, in the third step, it chooses the worst action of another sensor taking the first two sensors into consideration. The process then repeats until there is no sensor left. In this way, the inequation $V(S') \geq V(\alpha_1, \alpha_2, ..., \alpha_k) + \sum_{i=k+1}^{n} V(s_i)$ is more likely to happen as $V(\alpha_1, \alpha_2, ..., \alpha_k)$ is likely to be small.

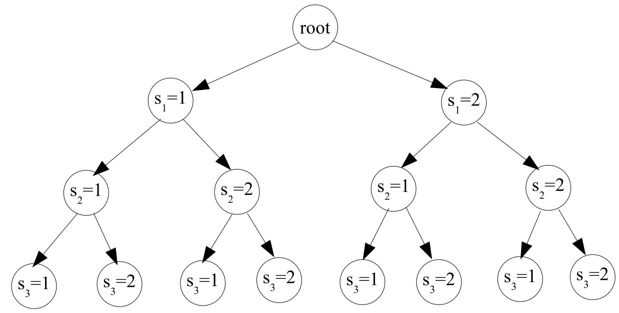

Figure 3: An example search tree with $n = 3$ and $k_i = 2$, for all $1 \leq i \leq n$.

Algorithm 2

1. *Search all $\langle s_1, s_2, ..., s_n\rangle$ in a depth-first search manner.*
2. *Suppose the current best system welfare is V^1 (initially V^1 would be the system welfare generated by the greedy algorithm). If $V^1 \geq V^u$, V^u is the upper bound of the value of any solution in the subtree $\langle\alpha_1, \alpha_2, ..., \alpha_k\rangle$ (i.e. $\langle s_1 = \alpha_1, s_2 = \alpha_2, ..., s_k = \alpha_k\rangle$), prune the whole subtree $\langle\alpha_1, \alpha_2, ..., \alpha_k\rangle$. The upper bound is derived from sub-additivity property of the valuation function as follows: for every $s_{k+1}, s_{k+2}, .., s_n$:*

$$\begin{aligned}&V(\alpha_1, \alpha_2, ..., \alpha_k, s_{k+1}, s_{k+2}, ..., s_n) \\ &\leq V(\alpha_1, \alpha_2, ..., \alpha_k, 0, 0, ..., 0) + \sum_{i=k+1}^{n} p_i\end{aligned}$$

   Thus if we have $V^1 \geq V(\alpha_1, \alpha_2, ..., \alpha_k, 0, 0, ..., 0) + \sum_{i=k+1}^{n} p_i$, the whole sub-tree $\langle\alpha_1, \alpha_2, ..., \alpha_k\rangle$ can be pruned safely.
3. *If we reach a leaf node in the tree, calculate its valuation and update the current best solution.*

Figure 4: The optimal coalition formation algorithm.

Experimental Results

This section outlines the experimental evaluation of our algorithms to see how they perform in reality. This is necessary because, for our polynomial algorithm, the theoretical analysis is in terms of worst-case, however, by doing an experimental analysis we can have a clearer idea of the typical performance; and for our optimal one, it is difficult to measure its effectiveness theoretically. Specifically, for the polynomial algorithm, we want to assess how close a typical solution is to the optimal compared to the worst-case bound, and for the optimal algorithm, we want to assess how effectively the search space is pruned. To this end, we next describe the experimental setup in subsection, and then present the evaluation results for the polynomial and optimal algorithms separately.

Experimental Setup

In order to generate generic problems on which to compare the performance of our two algorithms, we model the sensor

Algorithm 3

1. *Choose an agent with the biggest number of weak states.*
2. *Repeat step 1 until no agent with weak states is left.*
3. *For the remaining agents, carry out the following steps:*
   - *Each agent chooses its worst state taking only itself into consideration and calculates this worst outcome. That is, each agent i chooses its state s_i such that:*

     $$\sum_{j=1}^{m} v(\{i\}, t_j) - c_i \text{ is minimised}$$

     and calculates its worst outcome p_i. The agents then choose agent i_1 with the worst outcome:
     $p_{i_1} = \min_{i \in I} p_i$
   - *Each agent, except i_1, chooses its worst state taking only itself and i_1 into consideration and calculates this worst outcome. That is, each agent i chooses its worst state s_i such that:*

     $$\sum_{\substack{j=1 \\ C_j \subseteq \{i_1, i\}}}^{m} v(C_j, t_j) - c_{i_1} - c_i \text{ is minimised}$$

     and calculates its worst outcome p'_i. The agents then choose agent i_2 with the worst outcome:
     $p'_{i_2} = \min_{i \in I, i \neq i_1} p'_i$
   - *Repeat the above step until we reach the last agent.*

Figure 5: The tree structure selection process for the optimal coalition formation algorithm.

network described in figure 1. That is, we have n (ranging from 4 to 20 in the experiment) sensors, each with a fixed range and four distinct sensing states, randomly distributed within a unit area. Within this area are m (that has the same value as n in the experiment) targets, again randomly distributed. We assign a random sensing cost on the interval $[0, 1)$ to each sensor, and a random coalition value, again on the interval $[0, 1)$, to each coalition that contains a single sensor. We then use an iterative process to randomly assign the coalition values of all the larger coalition, whilst ensuring that these values satisfy our monotonicity and sub-additivity constraints. In this way, we calculate problem instances that are as general as possible, and thus, do not bias our results to a specific scenario.

Now, due to the demands of space, here we set n equal to m. Then for values of 4, 8, 12, 16 and 20 targets and sensors, we run the algorithms 200 times[1] and record the bound from the optimal (for the polynomial algorithm) and the percentage of pruned space (for the optimal algorithm).

[1] An ANOVA test showed that 200 iterations is sufficient for statistically significant results. For $\alpha = 0.05$, the p-value for the null hypothesis is > 0.05 in all the experiments with 5 samples. This shows that there is not a significant difference between the mean values and thus validates the null hypothesis.

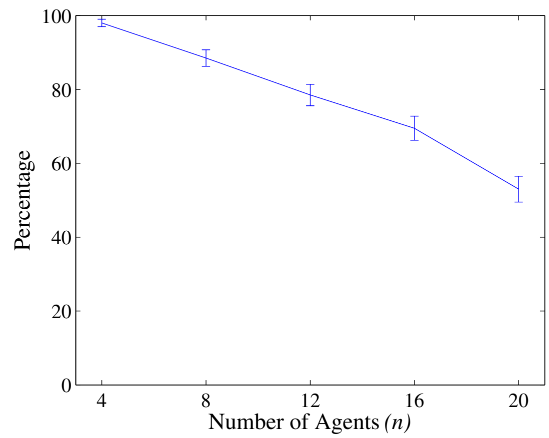

Figure 6: Percentage of searches that return the optimal value in the case of the polynomial algorithm.

Number of Sensors and Targets	Mean Bound	Std Dev
4	1.0028	0.024
8	1.0062	0.022
12	1.0071	0.023
16	1.0054	0.013
20	1.0078	0.012

Table 1: Polynomial algorithm – bound from the optimal.

The Polynomial Algorithm

The result for the polynomial algorithm is presented in table 1 and figure 6. As we can see from the table, all of the bounds are very close to 1. Specifically, the bound mean is always less than 1.01 and the standard deviation is always less than 0.03. This is close to the optimal and significantly lower than the theoretically proved bound which is n (i.e., 4 to 20 in this experiment). This suggests that in many practical cases, our algorithm performs significantly better than the theoretical proved worst-case analysis. Moreover, from figure 6, we see that when the number of sensors and targets is small, the greedy algorithm generates the optimal solution a significant percentage of the time (i.e. greater than 80% when $n = m = 8$). However, as the number of sensors and targets increase, the problem instances become more difficult to solve, and thus, this percentage decreases.

The Optimal Algorithm

The result for the optimal algorithm is presented in figure 7. This logarithmic plot shows the degree to which the branch-and-bound algorithm is able to exploit the known structure of the problem (i.e. monotonicity and sub-additivity) in order to be able to prune the search space. Note that when $n = m = 20$, the algorithm needs to typically only search 10^{-8} of the entire search space in order to calculate the optimal solution. As such, our algorithm hugely outperforms a naïve brute force approach.

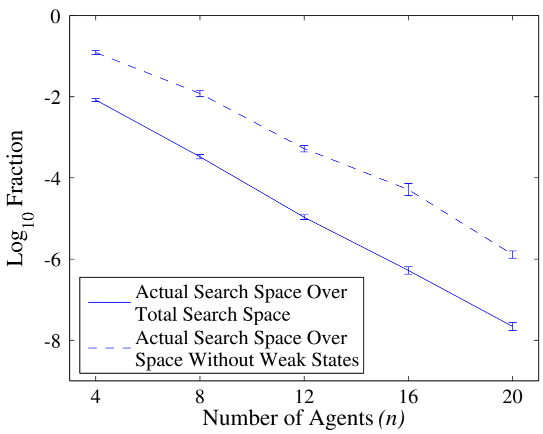

Figure 7: Comparison of various search spaces in the optimal algorithm.

Related Work

A number of algorithms have been developed for coalition formation, but in general these have not considered overlapping coalitions (Sandholm *et al.* 1999; Dang & Jennings 2004). However, the notion of overlapping coalitions was introduced by Shehory and Kraus in their seminal work on coalition formation for task allocation (Shehory & Kraus 1998; 1996). Here they developed a greedy algorithm for finding a solution to the overlapping coalitions problem that exhibited logarithmic bound. However, in contrast to our problem, they considered a specific block-world scenario in which the tasks had a precedence ordering and the agents had a capability vector (denoting the ability of the agents to perform tasks). As a result, the algorithm they develop is dissimilar to our polynomial algorithm. Moreover, they do not develop an optimal algorithm for the overlapping coalition formation process.

The application of coalition formation techniques to distributed sensor networks has also been investigated by a number of researchers. In (Sims, Goldman & Lesser 2003), a vehicle-tracking sensor network is modelled using disjoint coalitions formed via a negotiation process that results in a self-organising system. Similarly, in (Soh, Tsatsoulis & Sevay 2003), negotiation techniques are employed in order to form coalitions that track a target. However, these works focus on identifying and negotiating with potential coalition members since they operate in an incomplete information scenario where they are not aware about the existence and capabilities of other sensors. Our work on the other hand focuses on providing algorithms for the coalition formation process in a complete information environment.

Conclusions and Future Work

In this paper, we considered coalition formation for multi-sensor networks applied to wide-area surveillance. Specifically, we showed how this application leads to overlapping coalitions which exhibit sub-additivity and monotonicity, and we designed two novel coalition formation algorithms that exploit this particular structure. The first was an approximate and polynomial algorithm, with complexity $O(n^2m)$, that exhibited a calculated bound from the optimal of n. The second, which was optimal and based on a branch-and-bound heuristic, used a novel pruning procedure in order to reduce the number of searches required. We used empirical evaluations on randomly generated data-sets to show that the polynomial algorithm typically generated solutions much closer to the optimal than the theoretical bound, and to prove the effectiveness of our pruning procedure.

Future work will focus on employing these algorithms within dynamic environments where the values of the coalitions change with time, thereby causing the optimal coalition structure to vary. This will occur in our scenario as targets move in and out of the sensors' range of observation. We also plan to test these algorithms on real data from multi-sensor networks in order to further evaluate their performance in real-life scenarios.

Acknowledgement

This research was undertaken as part of the ARGUS II DARP. This 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 EPSRC, MoD and DTI.

References

Cormen, T., Leiserson, C. and Rivest, R. 1990. *Introduction to Algorithms*. MIT Press.

Dang, V. D. and Jennings, N. R. 2004. Generating coalition structures with finite bound from the optimal guarantees. In *Proc. 3rd Int. Conf. on Autonomous Agents and Multi-Agent Systems*, 564–571.

Dash, R. K., Rogers, A., Reece, S., Roberts, S. and Jennings, N. R. 2005. Constrained bandwidth allocation in multi-sensor information fusion: A mechanism design approach. In *Proc. 8th Int. Conf. on Information Fusion*.

Land, A. H. and Doig, A. G. 1960. An automatic method for solving discrete programming problems. *Econometrica* 28:497–520.

Lesser, V., Ortiz, C. and Tambe, M., eds. 2003. *Distributed Sensor Networks: A multiagent perspective*. Kluwer Publishing.

Reece, S. and Roberts, S. 2005. Robust, low-bandwidth, multi-vehicle mapping. In *Proc. 8th Int. Conf. on Information Fusion*.

Sandholm, T., Larson, K., Andersson, M., Shehory, O. and Tohme, F. 1999. Coalition structure generation with worst case guarantees. *Artificial Intelligence* 111(1-2):209–238.

Shehory, O. and Kraus, S. 1996. Formation of overlapping coalitions for precedence-ordered task-execution among autonomous agents. In *Proc. 2nd Int. Conf. on Multiagent Systems*, 330–337.

Shehory, O. and Kraus, S. 1998. Methods for task allocation via agent coalition formation. *Artificial Intelligence* 101(1–2):165–200.

Sims, M., Goldman, C. and Lesser, V. 2003. Self-organization through bottom-up coalition formation. In *Proc. 2nd Int. Conf. on Autonomous Agents and MultiAgent Systems*, 867–874.

Soh, L., Tsatsoulis, C. and Sevay, H. 2003. A satisficing, negotiated, and learning coalition formation architecture. In Ortiz, C., Lesser, V. and Tambe, M. eds., *Distributed Sensor Networks: A Multiagent Perspective*. Kluwer.