Agent-mediated Multi-step Optimization for Resource Allocation in Distributed Sensor Networks

Bo An, Victor Lesser, David Westbrook
Department of Computer Science
University of Massachusetts, Amherst, USA
{ban,lesser,westy}@cs.umass.edu

Michael Zink
Dept. of Electrical and Computer Engineering
University of Massachusetts, Amherst, USA
zink@ecs.umass.edu

ABSTRACT

Distributed collaborative adaptive sensing (DCAS) of the atmosphere is a new paradigm for detecting and predicting hazardous weather using a large dense network of short-range, low-powered radars to sense the lowest few kilometers of the earths atmosphere. In DCAS, radars are controlled by a collection of Meteorological Command and Control (MC&C) agents that instruct where to s-can based on emerging weather conditions. Within this context, this work concentrates on designing efficient approaches for allocating sensing resources to cope with restricted real-time requirements and limited computational resources. We have developed a new approach based on explicit goals that can span multiple system heartbeats. This allows us to reason ahead about sensor allocations based on expected requirements of goals as they project forward in time. Each goal explicitly specifies end-users' preferences as well as a prediction of how a phenomena will move. We use a genetic algorithm to generate scanning strategies of each single M-C&C and a distributed negotiation model to coordinate multiple MC&Cs' scanning strategies over multiple heartbeats. Simulation results show that as compared to simpler variants of our approach, the proposed distributed model achieved the highest social welfare. Our approach also has exhibited similarly very good performance in an operational radar testbed that is deployed in Oklahoma to observe severe weather events.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems

General Terms

Design, Experimentation, Performance

Keywords

Multi-agent systems, sensor networks, coordination, negotiation

1. INTRODUCTION

Over the last 6 years we have been developing and deploying a new paradigm called collaborative adaptive sensing of the atmosphere (CASA) for detecting and predicting hazardous weather [5, 15]. This new paradigm is achieved through a *distributed*, *collaborative*, *adaptive sensing* (DCAS) architecture. Distributed refers to the use of large numbers of small radars, whose range is short enough to see close to the ground in spite of the Earth's curvature and to avoid resolution degradation caused by radar beam spreading. Collaborative operation refers to the coordination of the beams from multiple radars to cover the blind, attenuated or cluttered regions of their neighbors and to simultaneously view the same region in space (when advantageous), thus achieving greater sensitivity, precision, and resolution than possible with a single radar. Adaptive refers to the ability of these radars and their associated computing and communications infrastructure to dynamically reconfigure in response to changing weather conditions and end-user needs. The principal components of a CASA DCAS system includes the sensors (radars); algorithms that detect, track, and predict meteorological hazards; interfaces that enable end-users to access and interact with the system; storage; and an underlying substrate of distributed computation that dynamically processes sensed data and manages system resources. At the heart of a DCAS system is its Meteorological Command and Control (MC&C) that perform-s the system's main control loop - ingesting data from the remote radars, identifying meteorological features in this data, reporting features to end-users, and determining each radar's future scan s-trategy based on detected features and end-user requirements.

MC&Cs' resource allocation problem of deciding radars' scan strategies is challenging due to a number of reasons. First, DCAS is an end-user driven approach and different users (e.g., National Weather Service (NWS) whose role is to issue severe weather watches and warnings, regional Emergency Managers (EMs) whose role is to alert the public about weather hazards and to coordinate first responders) have different data collection preferences and needs. Second, in DCAS, adaptive radars are controlled by a collection of MC&Cs and each MC&C tries to find the best scan strategy for the set of radars it controls. While such a distributed model brings some good properties (e.g., robustness, scalability), the problem of coordinating scan strategies of multiple MC&Cs arises as radars belonging to different MC&Cs may have an overlapping region. In certain situations, it is advantageous to have two or more radars focus their scans on overlapping regions in the atmosphere to provide accurate estimation of wind velocity vectors. In some other situations, a single radar's scanning can provide very high quality data and coordination can allow other radars to s-can other meteorological features. Third, DCAS is a real-time system and radars must be re-tasked by the MC&Cs every 60 seconds, which defines the system heartbeat interval [5, 15]. Therefore, the optimization for allocating radar resources should be completed in less than 60 seconds. Furthermore, the strategy space of each radar is infinite since each scan action can be represented by a region in the atmosphere.

In the previous resource allocation model [3], all the MC&Cs are myopically optimizing every "single" heartbeat's utilities without explicitly taking into account end-users' various needs over

Cite as: Agent-mediated Multi-step Optimization for Resource Allocation in Distributed Sensor Networks, Bo An, Victor Lesser, David Westbrook and Michael Zink, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems – Innovative Applications Track (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. XXX-XXX.

multiple heartbeats. It turns out this model leads to poor performance since subsequent changes in the environment may make these myopic decisions not appropriate anymore. For instance, an MC&C may repeatedly scan a high utility phenomenon (no matter how many times it has been scanned before) and thus miss some less important phenomena. Furthermore, a user's data requirement can be satisfied in different ways over multiple heartbeat intervals, which implies that it needs to search over multiple system heartbeats to find radars' optimal scanning strategies and address potential conflicts of available resources (sensing, computation, and bandwidth) over multiple heartbeats. In addition, predictions about future events [9] are useless in this optimization framework.

The focus of this paper is investigating the practicality of applying real-time distributed multi-step optimization approaches in a real application involving complex resource allocation. We found that a real-time distributed multi-step optimization approach is feasible and it contributes to better performance. This paper proposes a novel distributed resource allocation approach to address diverse user preferences over multiple heartbeats. We introduce the concept "goal" (or constraint) to represent end-users' preferences on radars' scan strategies. Each goal specifies the region of a phenomenon over multiple heartbeats and how well a user's preference is satisfied given radars' scan strategies over multiple heartbeats. Then the resource allocation problem can be formulated as a continuous time constraint optimization problem. The goal based formulation allows us to reason ahead about allocations based on expected requirements of goals over multiple heartbeats and prediction about future weather phenomena. Given that the strategy space of each radar is continuous and the real-time requirement, it is impractical to exhaustively search all the possible strategies. Alternatively, each MC&C finds approximate local optimal solutions employing a genetic algorithm. Different strategies are mapped into chromosomes and genetic operators like mutation, selection, and crossover are employed. A distributed asynchronous negotiation model is used to coordinate the scan strategies of multiple MC&Cs. Each MC&C always notifies its current multi-heartbeat strategy to its neighbor MC&Cs. Based on the current strategies of its neighbor MC&Cs, an MC&C proposes to change its strategy and decides whether to make the change based on the marginal utilities of its neighbors' strategy changes. This asynchronous negotiation continues until the heartbeat deadline approaches. Simulation results show that as compared to other mechanisms, the proposed distributed model achieved the highest social welfare. We have also verified the performance of our approach in the operational radar testbed deployed in Oklahoma while it was responding to actual severe weather events. These empirical results mirror the positive results achieved in our simulation studies.

The remainder of this paper is organized as follows. Section 2 discusses related work. We next formalize the resource allocation problem in Section 3. We then discuss the genetic algorithm for finding each MC&C's local optimal strategies. The distributed negotiation model is presented in Section 5. Section 6 reports simulation results. Section 7 discusses the performance of our approach in the real sensor system and Section 8 concludes this paper.

2. RELATED WORK

The development of decentralized optimization and coordination techniques to achieve good system-wide performance is a fundamental challenge for practical distributed sensor networks, which mainly comes from various constraints, e.g., realtime response, limited communication and computational resources. While multi-agent systems community has developed a variety of techniques for distributed resource allocation in sensor networks [4, 8], these approaches cannot be directly applied to our special domain with complex user preferences.

The problem of decentralized coordination can be formulated as a distributed constraint optimization problem (DCOP), which makes it possible for us to use a wide range of existing algorithms for DCOP, e.g., ADOPT [6]. However, these complete algorithms cannot be directly applied to problem due to their limitations such as high computational complexity and large size of exchanged messages. Furthermore, these algorithms are for one-step optimization but our problem is a continuous optimization problem. While there have been numerous approximate stochastic algorithms based on entirely local computation for solving DCOPs [14], these algorithms often converge to poor-quality solutions because agents typically communicate only their preferred state, failing to explicitly communicate utility information [8]. Max-sum algorithm has recently been applied to the sensor network domain (e.g., [12]), our recent study [2] showed that the max-sum algorithm did not outperform the approach in [3] and had a much worse performance when there were more overlapping radars.

Negotiation has been used in distributed sensor networks in the past; however, previous techniques are not entirely appropriate for our setting. In the argumentation-based approach [11], an initiator attempts to recruit other sensors to scan a specific task. In our domain, a per goal negotiation would not be feasible based on time limitations. Contract-net based negotiation schemes [10] in which agents make bids based on utility calculations face similar limitations. If the contract net protocol is adopted, every time an MC&C's neighbor changes its scan strategy, that MC&C must perform potentially as many optimizations for marginal utility calculations as the size of the powerset of the boundary goals belonging to it. The similar problem exists while adopting combinatorial auctions [1]. The negotiation model for single step optimizations for the DCAS system [3] fails to capture users' preferences over multiple heartbeats and accordingly, may result in low social welfare due to lack of reasoning about future actions. In addition, the synchronous negotiation protocol in [3] may have bad performance due to its lack of concurrency in real time optimization.

3. PROBLEM FORMULATION

This section formalizes the problem of optimizing resource allocation which has *observed phenomena* as its input and *scan commands* as its output. The following components are involved in solving the meteorological control problem: goal generation, local optimization that generates scan commands for each MC&C's radars, and negotiation which coordinates MC&Cs' scan actions.

3.1 Goal Generation

In the current design, the DCAS system dynamically adapts radar scans at 60 second intervals to sense the evolving weather and disseminates information to users based on their changing and diverse preferences for data. An NWS forecaster may analyze the vertical structure of a storm to determine whether to issue a warning by viewing a sector scan at multiple elevations, while an emergency manager may require two radars to collaborate in order to pinpoint the location of the most intense part of a storm for spotter deployment, and a researcher may require 360 degree scans at all elevations to initialize a numerical weather prediction model. These diverse information preferences require different radar scan strategies. We use *scan goals* to formulate diverse user preferences and phenomena regions over multiple heartbeats. A goal g specifies:

- Generation time $T_s(g)$.
- Deadline $T_e(g)$. There could be a goal existing for only one heartbeat, i.e., $T_e(g) = T_s(g)$. It is also possible that $T_e(g) -$

$T_s(g) > 0$, i.e., the satisfaction of the goal may involve scan actions over multiple heartbeats.

- Scan area(s). A goal g is either to find new phenomena or to find more details of an existing known phenomenon. For the former case (e.g., 360° scan), the scan area will not change over time. For the latter case, as a phenomenon moves over time, the scan areas at different heartbeats may be different which depends on the moving speed of a phenomenon and how its shape is expected to change over time. Let $A(g,t)$ denote the scan area of goal t at heartbeat $t \in [T_s(g), T_e(g)]$ and such information can be gained by prediction [9]. A goal may be updated later due to imprecise prediction. An area A (or part of it) may fall within the coverage of a radar r, i.e., $\Psi(A,r) = \text{true}$.
- Utility calculation function $U_g(s_{\mathcal{R}}^{T_s(g)\to T_e(g)})$ which defines how well the goal is satisfied based on radars $\mathcal{R}$'s scan actions $s_{\mathcal{R}}^{T_s(g)\to T_e(g)}$ from heartbeat $T_s(g)$ to $T_e(g)$.

In summary, goals specify 1) in what manner different kinds of weather phenomena should be scanned by radars and 2) how well different user groups are satisfied given radars' scan strategies. A goal generation rule specifies when the rule is triggered to generate a new goal and how to set the properties of the new goal. A simple example of goal generation rules is that each radar needs to do a 360° scan every 5 minutes (heartbeats).

Note that each MC&C generates goals individually. It's possible that two MC&Cs generate goals for the same phenomena which is located on the overlapping area of the two MC&Cs. In such cases, coordination mechanisms (Section 5) are used to resolve such conflicts. When a goal is generated to find the details of a known phenomenon, the goal will also specify its "regions" in the future based on the prediction about the phenomenon's moving speed and change of its shape. Therefore, an MC&C may also update the properties of an existing goal based on its new observations. This update is important as prediction made at goal generation time may not be accurate enough.

3.2 Goal Satisfaction

Let the set of MC&Cs be $\mathcal{M} = \{M_1, \ldots, M_{|\mathcal{M}|}\}$ and the set of radars be $\{\mathcal{R}_1, \ldots, \mathcal{R}_{|\mathcal{M}|}\}$, where $\mathcal{R}_i$ is the set of radars controlled by MC&C M_i and $\mathcal{R}_i \cap \mathcal{R}_j = \emptyset$. Each radar has its coverage area and the coverage area of an MC&C includes the coverage areas of all its radars. MC&C M_i is a neighbor of M_j if their coverage areas overlap. Let $\mathcal{NM}_i$ denote the set of neighbor MC&Cs of M_i. Let $\mathcal{G}_i^t$ be the set of goals generated for M_i at the beginning of heartbeat t, i.e., for any $g \in \mathcal{G}_i^t$, $T_s(g) = t$. Accordingly, goals generated by all MC&Cs at heartbeat t is $\mathcal{G}^t = \cup_{M_i \in \mathcal{M}} \mathcal{G}_i^t$. Let $\mathcal{G}_i^{t\to t'}$ be the set of goals generated for MC&C M_i from heartbeat t to heartbeat t', i.e., $\mathcal{G}_i^{t\to t'} = \cup_{t \le t'' \le t'} \mathcal{G}_i^{t''}$. A goal g is *active* at time t if $t \in [T_s(g), T_e(g)]$. Let $\mathcal{AG}_i^t$ be the set of *active* goals of MC&C M_i at heartbeat t, i.e., $\mathcal{AG}_i^t = \{g | g \in \mathcal{G}_i^{0\to t}, T_e(g) \le t\}$. Accordingly, active goals for all MC&Cs at heartbeat t is $\mathcal{AG}^t = \cup_{M_i \in \mathcal{M}} \mathcal{AG}_i^t$. Out of the set of goals in $\mathcal{AG}_i^t$, some are *boundary* goals $\mathcal{BG}_i^t$. A goal $g \in \mathcal{AG}_i^t$ is a boundary goal if there exits a radar r' belonging to another MC&C and one of goal g's scan area from time t could be partially covered by r', i.e., $\Psi(A(g,t'), r') = \text{true}$ for some $r' \in \mathcal{R}_j$ and $t \le t' \le T_e(g)$.

We assume that the set of end-users are $\mathcal{K}$. Let $w_k(g)$ be the weight associated with user $k \in \mathcal{K}$ for goal g. The user weight $w_k(g)$ reflects 1) the relative priority of user k with respect to other users and 2) the importance of goal g from user k's perspective. The values of $w_k(g)$ are set by high-level system user policies. A radar's scan action (strategy) can be defined to be the start and end angles of the sector to be scanned by an individual radar for a fixed interval of time (a heartbeat). Utility evaluation of a goal depends on both scan quality and weight. Quality measures how well an area is scanned, with quality depending on the amount of time a radar spends sampling a voxel in space, the degree to which an area is scanned in its (spatial) entirety, and the number of radars observing an area.

Quality function: The quality $Q(A, s_r)$ of scanning an area A using scan action s_r by a single radar r can be defined as

$$Q(A, s_r) = F_c(c(A, s_r)) \times \left[\beta F_d(d(r,A)) + (1-\beta) F_w\left(\frac{wd(s_r)}{360}\right)\right]$$

where $wd(s_r)$ is the size of sector s_r, $a(r,A)$ is the minimal angle that would allow r to cover A, $c(A, s_r) = \frac{wd(s_r)}{a(r,A)}$ is the coverage of A by s_r, $h(r,A)$ is the distance from r to geometric center of A, $h_{max}(r)$ is the range of radar r, $d(r,A) = \frac{h(r,A)}{h_{max}(r)}$ is the normalized distance from r to A, and β is a tunable parameter. F_c captures the effect on quality due to the percentage of the area covered. F_w captures the effect of radar rotation speed on quality. F_d captures the effects of the distance from the radar to the geometrical center of the phenomenon area.

A scan area may be scanned by more than one radar in the same heartbeat. $Q(A, s_{\mathcal{R}}^t)$ is the maximum quality obtained for scan area A over a set of radars $\mathcal{R}$ and their scan actions $s_{\mathcal{R}}^t$ at time t. If the phenomenon corresponding to the scan area A is a *pinpointing* phenomenon, $Q(A, s_{\mathcal{R}_i}^t)$ is defined as $Q(A, s_{\mathcal{R}}^t) = \sum_{r \in \mathcal{R}} Q(A, s_r^t)$ where s_r^t is the scan action for radar r at time t. Otherwise, $Q(A, s_{\mathcal{R}}^t) = \max_{r \in \mathcal{R}} Q(A, s_r^t)$.

We can get user k's utility $U_g(k, s_{\mathcal{R}}^t)$ of satisfying the goal g given the scan actions $s_{\mathcal{R}}^t$ by combining the weight component and the quality component. Formally

$$U_g(k, s_{\mathcal{R}}^t) = \begin{cases} \delta^{(t - T_s(g))} w_k(g) Q(A(g,t), s_{\mathcal{R}}^t) & \text{if } T_s(g) \le t \le T_e(g) \\ 0 & \text{otherwise} \end{cases}$$

where $\delta \in (0.1]$ is a discount factor reflecting a user's eagerness of scanning a phenomenon earlier.

Let $U_g(k, s_{\mathcal{R}}^{t\to t'})$ be user k's utility of satisfying goal g based on a series of scan actions $s_{\mathcal{R}}^{t\to t'} = \{s_{\mathcal{R}}^t, \ldots, s_{\mathcal{R}}^{t'}\}$ from t to t'. There are multiple ways of defining $U_g(k, s_{\mathcal{R}}^{t\to t'})$, e.g., $\max_{t \le q \le t'} U_g(k, s_{\mathcal{R}}^q)$, $\max_{t \le q < t'} (U_g(k, s_{\mathcal{R}}^q) + U_g(k, s_{\mathcal{R}}^{q+1}))$, $\max_{t \le p < q \le t'} (U_g(k, s_{\mathcal{R}}^p) + U_g(k, s_{\mathcal{R}}^{q+1}))$, or $\sum_{t \le q \le t'} U_g(k, s_{\mathcal{R}}^q)$. Given actions $s_{\mathcal{R}}^{t\to t'}$, the aggregate utility $U_g(s_{\mathcal{R}}^{t\to t'})$ for satisfying a goal g is the sum $\sum_{k \in \mathcal{K}} U_g(k, s_{\mathcal{R}}^{t\to t'})$ of utilities of all users.

3.3 Formulation of the Optimization Problem

The objective of the optimization is to satisfy the set of goals $\mathcal{G}^0, \mathcal{G}^1, \ldots, \mathcal{G}^\infty$. At heartbeat t, MC&Cs need to determine optimal radar scanning actions at t and later heartbeats for active goals $\mathcal{AG}^t$. However, limited computational resources preclude that we could compute the optimal actions from now to the infinite future. Instead, we adopt the *receding horizon control* principle by focusing on the optimal actions $s_{\mathcal{R}}^{t\to t+l-1}$ in heartbeats of length l:

$$\arg \max_{s_{\mathcal{R}}^{t\to t+l-1}} \sum_{g \in \mathcal{AG}^t} U_g(s_{\mathcal{R}}^{0\to t-1} \cup s_{\mathcal{R}}^{t\to t+l-1})$$

This formulation is in some sense "myopic" as, in fact, MC&Cs need to consider what's going to happen in the future while deciding "optimal" actions at heartbeat t. As it is not possible to obtain perfect information about future states, a guaranteed optimal solution is not possible to obtain (even neglecting the computational intractability nature of the problem at hand). Although the optimization process at heartbeat t will output a schedule over multiple heartbeats, only the scan strategies at time t will be executed by radars. At time $t+1$, each MC&C updates its goal sets, possibly generates new goals, and runs the optimization algorithm again.

4. LOCAL OPTIMIZATION OF EACH M-C&C

This section discusses how an MC&C M_i searches scanning actions for its radars $\mathcal{R}_i$ given the set of active goals $\mathcal{AG}_i^t$ at heartbeat t. The optimization problem of MC&C M_i at time t is to find the best scan strategy $s_{\mathcal{R}_i}^{t\to t+l-1}$ for its radars. Formally,

$$\arg\max_{s_{\mathcal{R}_i}^{t\to t+l-1}} \sum_{g\in\mathcal{AG}_i^t} U_g(s_{\mathcal{R}_i}^{t\to t+l-1} \cup s_{\mathcal{R}_i}^{0\to t-1} \cup s_{\mathcal{R}_{-i}}^{0\to t+l-1})$$

where $s_{\mathcal{R}_{-i}}^{0\to t+l-1}$ are the strategies of M_i's neighbor MC&Cs, which can be known to M_i through information exchange - negotiation.

The search depth l should be no larger than $\max_{g\in\mathcal{AG}_i^t} T_e(g)$ and setting search depth l involves a number of tradeoffs. With a larger search depth l, M_i has a larger space to coordinate the scan strategies of its radars. However, as only the scan strategy at time t will be executed, the optimal strategy $s_{\mathcal{R}_i}^{t\to t+l-1}$ found at time t may be not optimal in practice. Furthermore, the computational complexity of searching for optimal strategies increases with the search depth l. In addition, the prediction of the movement of observed phenomena could be inaccurate. When search depth l is large, the propagation of such inaccuracy could lead to poor performance of the scan strategies.

Since the strategy space of each radar is continuous, we first discretize the radar's strategy space such that the start and end angles of each strategy can only be in $\{0, 5, 10, \ldots, 360\}$.[1] Then for each goal $g\in\mathcal{AG}_i^t$ and each radar $r\in\mathcal{R}_i$ such that $\Psi(A(g,t'), r) = \mathrm{true}$ at $t \le t' \le T_e(g)$, generate the minimum sector that can cover the region $A(g,t')$ and add the sector to the candidate strategy set $\mathcal{S}_r^{t'}$ of radar r. If $\mathcal{S}_r^{t'}$ contains more than λ strategies, combine two randomly selected strategies into one strategy and this process continues until $|\mathcal{S}_r^{t'}| = \lambda$. The maximum size of M_i's strategy space is $|\mathcal{R}_i|^{\lambda^l}$. For ease of analysis, we assume that each strategy in $\mathcal{S}_r^{t'}$ has an ID ranging from 0 to $|\mathcal{S}_r^{t'}|-1$. Similarly, we give each radar $r\in\mathcal{R}_i$ an ID ranging from 0 to $|\mathcal{R}_i|-1$.

The complexity of the optimization problem precludes an M-C&C from using an exhaustive search to find its optimal solution. Alternatively, we use a genetic algorithm (GA) to search the (nearly) best solution. The GA generates a sequence of populations as the outcome of a search method. The individuals of the population are scan strategies over multiple heartbeats. Each strategy combination can be represented as a matrix of size $|\mathcal{R}_i| \times l$ in which column j represents radars' scanning strategies at heartbeat $t+j$ and row i represents radar i's scanning strategies from heartbeat t to heartbeat $t+l-1$. Let the matrix for a strategy combination be X. Then $x_{i,j}$ represents radar i's scanning strategies from heartbeat $t+j$ and it follows that $x_{i,j} \in [0, |\mathcal{S}_i^{t+j}|-1]$.

An individual's fitness value is determined by the utility of all the goals $\mathcal{AG}_i^t$ while all radars take strategies of the individual. The evolution starts from a population of randomly generated individuals. In each generation, operators selection, crossover (recombining existing genetic materials in new ways) and mutation (introducing new genetic materials by random modifications) are used to form a new population. The new population is then used in the next iteration of the algorithm. The algorithm terminates when the local optimization deadline τ (e.g., 5 seconds) has been reached, or the population is stable (e.g., 95% of the individuals have the same highest fitness value). When the genetic algorithm terminates, the chromosome that has the highest fitness is extracted and the decoded strategies are the best strategies for the MC&C.

[1]This does not have a substantial impact on the system since we always scan a little wider than the edges of a phenomena anyway.

Algorithm 1: The Negotiation Algorithm for MC&C M_i

```
Let Θ ∈ {wstrategy, wproposal} represent the status of MC&C M_i.
Let function GetTime() return the current time.
Let Ω_str/Ω_move be the queue to store other MC&Cs' strategy
update/proposals.
Initialization:
    a). Send goal set AG_i^t to its neighbor MC&Cs (Ω_str = NM_i);
    b). Run the genetic algorithm and get optimal scanning strategies s_{R_i};
    c). Send s_{R_i} to all neighbor MC&Cs;
    d). Set Θ = wstrategy and nowt = GetTime();
while optimization deadline has not expired do
    if Θ=wstrategy and Ω_str ≠ ∅ then
        if (GetTime() − nowt) > ξ or Ω_str = NM_i then
            Run the genetic algorithm and get new optimal strategies s'_{R_i};
            if MC&C M_i can gain positive marginal utility by using s'_{R_i}
            then
                Send s'_{R_i} with its marginal utility to all neighbor MC&Cs;
                Set Θ = wproposal, Ω_move = ∅,
                nowt = GetTime();
    else if Θ=wproposal and (GetTime() − nowt) > ξ then
        if The marginal utility of MC&C M_i by using s'_{R_i} is higher than its
        neighbor MC&Cs' marginal utility then
            Set s_{R_i} = s'_{R_i}, Ω_str = ∅;
            Send s'_{R_i} to all neighbor MC&Cs NM_i;
        Set Θ = wstrategy, nowt = GetTime();
```

When each MC&C runs the local optimization algorithm separately resulting in a scan strategy based on its local (partial) view of the physical space, efficiency loss may occur. One such source of quality degradation is the loss of the ability to cooperatively scan pinpointing phenomena on boundaries, which can be solved by coordinating scans between MC&Cs and sharing resulting raw data. Another source of lessened quality are redundant scans which can be alleviated by allowing MC&Cs to share abstract level information regarding goals located in boundaries. The limitations of the fully distributed optimization lead us to study the coordination problem of distributed MC&Cs.

5. NEGOTIATION BASED COORDINATION

This section extends the negotiation model in [3] to accommodate 1) user's complex preferences over multiple heartbeats and 2) the need of concurrency during negotiation. In [3], all the MC&Cs ignore end users' preferences over multiple heartbeats and are only maximizing the social welfare of a "single" heartbeat. Accordingly, the model in [3] may lead to poor performance due to its lack of reasoning ahead. In [3], MC&Cs conduct synchronous negotiation. That is, after an MC&C makes proposals to its neighbors, it will respond to its neighbors only after it has received all responses from its neighbors. While the synchronous protocol can guarantee that the social welfare will improve after each round of negotiation, it may be unreasonable for an MC&C to wait for responses from its neighbors given the real time constraints and bounded computational resources.

Algorithm. 1 shows how the distributed negotiation is conducted between MC&Cs at heartbeat t. For a boundary phenomenon, it is possible that one MC&C observes it while another MC&C fails to discover it. Before MC&Cs begin the main stages of negotiation, each MC&C communicates with its neighbors MC&Cs to make sure its boundary goals are also in the goal sets of other MC&Cs.

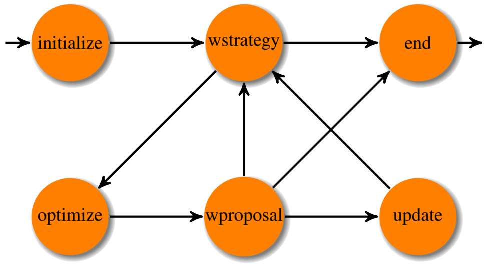

Figure 1: Finite state machine for distributed negotiation

Then each MC&C runs its local optimization algorithm (Section 4) to generate its initial strategy over multiple heartbeats. Next each MC&C shares its initial configuration with its neighbor MC&Cs. Then the main stages of negotiation starts. The negotiation stops when the heartbeat deadline (60 seconds) approaches.

Since an MC&C's optimal strategy depends on its neighbor MC&Cs' scan strategies. After one neighbor MC&C changes its strategy, M_i's optimal strategy $s_{\mathcal{R}_i}$ may be not optimal anymore. Thus, M_i can run its local optimization algorithm again to find its new optimal scan strategy $s'_{\mathcal{R}_i}$. M_i's marginal utility $UM_i(s'_{\mathcal{R}_i}, s_{\mathcal{R}_i})$ when switching to strategy $s'_{\mathcal{R}_i}$ is $\sum_{g \in \mathcal{AG}_i^t} U_g(s'_{\mathcal{R}_i} \cup s_{\mathcal{R}_i}^{0 \to t-1} \cup s_{\mathcal{R}_{-i}}^{0 \to t+l-1}) - \sum_{g \in \mathcal{AG}_i^t} U_g(s_{\mathcal{R}_i} \cup s_{\mathcal{R}_i}^{0 \to t-1} \cup s_{\mathcal{R}_{-i}}^{0 \to t+l-1})$.

If $UM_i(s'_{\mathcal{R}_i}, s_{\mathcal{R}_i}) > 0$, M_i may choose to use strategy $s'_{\mathcal{R}_i}$. Since it is possible that other MC&Cs change their strategies simultaneously, M_i's new strategy $s'_{\mathcal{R}_i}$ may be not optimal any more since the optimality of $s'_{\mathcal{R}_i}$ is based on the assumption that M_i's neighbor MC&Cs don't change their strategies. To overcome the efficiency loss due to concurrency, a synchronization mechanism is used: An MC&C first proposes a strategy move by reporting its new strategy as well as its marginal utility to its neighbors, and then it changes its strategy if and only if its marginal utility is higher than the marginal utilities of its neighbor MC&Cs whose proposed *moves* are in *conflict* with the MC&C's proposed move. Since MC&Cs operate in real-time, it is possible that M_i fails to receive the proposal from one of its neighbor MC&Cs or it has to wait for a long time before receiving all proposals. To improve concurrency, we introduce a waiting deadline $\xi > 0$. MC&C M_i will decide whether to make a move after the waiting deadline expires.

Assume that M_i received a message from M_j indicating that M_j will change its strategy from $s_{\mathcal{R}_j}$ to $s'_{\mathcal{R}_j}$. M_j's move $s'_{\mathcal{R}_j}$ is in conflict with M_i's move $s'_{\mathcal{R}_i}$ if both moves will change the utility of some active goals $\mathcal{G} \subseteq \mathcal{AG}_i^t$. Let $\mathcal{NM}_i(s'_{\mathcal{R}_i})$ be the set of neighbor MC&Cs whose proposed moves are in *conflict* with the M_i's proposed move $s'_{\mathcal{R}_i}$. If the marginal utility increase of M_i's proposal is higher than the marginal utility of any MC&C in $\mathcal{NM}_i(s'_{\mathcal{R}_i})$, MC&C M_i will change its strategy to $s'_{\mathcal{R}_i}$. The complexity of this conflict check for each MC&C is $\mathcal{O}(|\mathcal{M}|)$. Note that it is also possible that the utility of a goal set will increase when multiple MC&Cs change their strategies simultaneously, no matter whether their moves are in conflict with each other. Since an MC&C's changing its strategy will affect the utilities of its neighbor MC&Cs, an MC&C's making the optimal decision of whether to switch to its new strategy $s'_{\mathcal{R}_i}$ may depend on other MC&Cs' choice of whether to change to their strategies.

DEFINITION 1. *(Move selection) Assume that MC&Cs' current strategies are* $s_{\mathcal{R}_1}, \ldots, s_{\mathcal{R}_{|\mathcal{M}|}}$*, respectively. Assume that MC&Cs are proposing to use new strategies* $s'_{\mathcal{R}_1}, \ldots, s'_{\mathcal{R}_{|\mathcal{M}|}}$*, respectively. The move selection problem is to find out the set of moves to maximize the social welfare.*

THEOREM 2. *The move selection problem is* $\mathcal{NP}$*-hard.*

The theorem's proof is a straightforward reduction from the maximum matching problem (omitted due to space limitations). Considering the high complexity of finding MC&Cs' optimal decisions of changing their strategies and the dynamic feature of the system, we adopt the above conflict check approach which has a low complexity since each MC&C only needs to consider the marginal utilities of its neighbor MC&Cs.

Figure 1 shows an MC&C's finite state machine for the distributed negotiation protocol. After receiving data from radars, M_i runs the local optimization algorithm to find its initial strategy. After it sends its strategy to its neighbor MC&Cs, M_i is in the state wstrategy, which implies that M_i is waiting for other MC&Cs to report their current strategies. After M_i has received strategies from all its neighbors or its waiting deadline ξ has reached, it computes its new optimal strategy and notifies its neighbor MC&Cs. Then its state is wproposal which implies that M_i has sent out its move proposal and is waiting for other MC&Cs to report their move proposals. If M_i's marginal utility is higher than the other marginal utilities of conflicting move proposals it has received within the waiting time, it will change its strategy and notify its neighbor MC&Cs. Then its status will be changed to wstrategy. During negotiation, after M_i decides whether to make a move, it will wait for other MC&Cs' strategy update. If the optimization deadline is reached, the state is end and M_i sends out its current scan commands to all the radars under its control.

There are several important control parameters in our approach and we set the values for those control parameters through experimental tuning. MC&C M_i needs to decide its search depth for local multi-step optimization. One obvious rule is that the search depth l should be no larger than $\max_{g \in \mathcal{AG}_i^t} T_e(g)$. Although an MC&C has a better chance to coordinate its future actions with a larger search depth, having a large search depth brings several drawbacks. First, the MC&C's strategy space increases exponentially with l. After generating a strategy over multiple heartbeats at heartbeat t, the MC&C will run the optimization algorithm again at heartbeat $t+1$. That is, the strategy for future heartbeats generated at time t may be abandoned later. Furthermore, as each MC&C has imperfect knowledge about future events due to inaccurate prediction and about the strategies of other MC&Cs, the generated "optimal" strategy over multiple heartbeats may not be optimal in practice. We used a heuristic to decide the search depth for each MC&C by considering a goal's expected existence time. A goal $g \in \mathcal{AG}_i^t$ will exist for $T_e(g) - t + 1$ heartbeats starting from heartbeat t. The average existing time $\sum_{g \in \mathcal{AG}_i^t} (T_e(g) - t + 1)/|\mathcal{AG}_i^t|$ of active goals $\mathcal{AG}_i^t$ is chosen as the search depth. Simulation results show that the heuristic achieved the highest utility compared to other arbitrary approaches (e.g., $l = 1$, $l = \max_{g \in \mathcal{AG}_i^t} T_e(g)$) for setting the search depth.

Two additionally important control parameters for each MC&C are the time τ to run its local optimization and the waiting time ξ during negotiation. With longer time, an MC&C can get a solution closer to the local optimal solution. However, an MC&C may has a short time for negotiation if it spends too much time in local optimization. During negotiation, an MC&C needs to decide how long to wait for the messages from its neighbor MC&Cs. With the increase of waiting time ξ, the negotiation is more synchronous since an MC&C will have more knowledge about its neighbor MC&Cs before making a decision. We found through experiments that it's always better to allocate 6 seconds for each local optimization. When $\tau \ll 6$ seconds, the local optimization solution has a low quality. If $\tau \gg 6$ seconds, there is not much time to do negotiation given 60 seconds heartbeat deadline. Furthermore, we found that it's always better to set $\xi \sim 4$ seconds. Therefore, MC&Cs may conduct 6 rounds of negotiation and we found that in most cases, negotiation converges (i.e., no MC&C can find a better strategy) in about 5 rounds of negotiation.

Our negotiation scheme has a number of features: 1) Each MC&C exchanges its scan plan (generated by local optimization) over multiple heartbeats with neighbor MC&Cs. 2) To reduce the utility loss due to concurrent strategy change, an MC&C changes its strategy if it has the highest marginal utility than that of its neighbor MC&Cs. 3) Negotiation is conducted asynchronously to increase concurrency of MC&Cs' strategy change. If we synchronize the negotiation protocol by setting a long waiting time, we can guarantee that the social welfare will monotonically increase with the ongoing negotiation as in [3]. We make tradeoffs between speeding up negotiation and guaranteeing monotonic increase of social welfare by setting the value of waiting deadline ξ by considering factors such as the communication delay distribution. When we set a long waiting deadline (i.e., there is no concurrency), the protocol is similar to the LID-JESP algorithm [7] which makes use of the distributed breakout algorithm (DBA) algorithm [13].

6. SIMULATION RESULTS

Evaluating the performance of the approach on the real radar system is difficult and complex. To better quantify the benefits of our approach, we turn to simulation results in more controlled settings.

6.1 Simulator

To determine how best to decentralize control, we have created an abstract simulation of the actual DCAS system. The simulator consists of a number of components. Radars are clustered into partitions, each of which has a single MC&C. Each MC&C has a feature repository where it stores information regarding phenomena in its spacial region, where each phenomenon represents a weather event. Goals are generated following goal generation rules given observed weather phenomena. The optimization function of each MC&C takes its scan goals and returns scans for each of its radars. The simulator additionally contains a function which abstractly simulates the mapping from physical events and scans of the radars to what the MC&C eventually sees as the result of those scans. Depending on the elevations scanned, the number of radars scanning, the type of phenomena, and the speed of scan, it assigns error values to the attributes of the phenomena within certain bounds. In this way, the MC&Cs do not see exactly what is there but rather something slightly off.

The parameters of each phenomenon (e.g., speed, density), each radar (e.g., radius), and each MC&C (e.g., the number of radars under control) reflect the current design of the real system. Phenomena may be either pinpointing or non-pinpointing. Goal generation and utility calculation in the simulator are the same as that in the real system. The radars have a range of approximately 30 kilometers and the optimization has to finish in 60 seconds. The communication delay between MC&Cs is based on the data gathered from the real system. The number of radars ranges between $[8, 100]$ and the number of radars controlled by each MC&C is ranged between $[4, 16]$. Each radar can have at most $\lambda = 8$ candidate strategies at any heartbeat which represent a wide range of strategies.

6.2 Benchmark approaches

Our *distributed negotiation (*$\mathbf{DN}$*)* model was compared with four other different approaches. Both the *centralized single-step optimization (*$\mathbf{CS}$*)* approach and *centralized multi-step optimization (*$\mathbf{CM}$*)* approach assume that there is a super MC&C controlling all the MC&Cs and the super MC&C runs the local optimization algorithm to find optimal strategies for all MC&Cs' radars. The only difference between $\mathbf{CS}$ and $\mathbf{CM}$ is that $\mathbf{CS}$ makes one heartbeat optimization while $\mathbf{CM}$ searches over multiple heartbeats. For the *fully distributed (*$\mathbf{FD}$*) approach*, MC&Cs optimize the utilities of their separate goal sets and don't communicate with each other. $\mathbf{FD}$ searches strategies over multiple heartbeats.

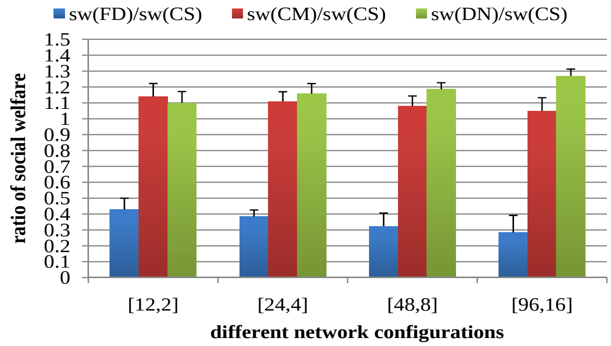

Figure 2: Ratio of social welfare and scale of the network.

6.3 Experimental settings

In our experiments, the simulator will model real phenomena generation and different approaches may generate different set of goals given their observations. Each approach will optimize its scan strategy based on its own goal set. Thus it is unfair to compare the performance of different approaches based on their own goal sets. Instead, we generate an *oracle* goal set based on the system's real phenomena. The social welfare of each approach is evaluated based on the actions generated by each approach and the set of oracle goals generated by the system. For an experiment, we run the system for multiple heartbeats (e.g., 200) and compute the average social welfare for each heartbeat, e.g., average social welfare $sw(\mathbf{DN})$ for the approach $\mathbf{DN}$.

An extensive amount of stochastic simulations was carried out for various resource allocation scenarios subjected to the following variables: 1) the scale of each MC&C, i.e., how many radars are controlled by an MC&C; 2) the density of phenomena, i.e., the frequency of new phenomena entering the radar network; 3) the speed of phenomena; and 4) the ratio of boundary goals. In the rest of this section, we report some representee simulation results.

6.4 Observations

6.4.1 Scale of the sensor network

On average, $\mathbf{DN}$ achieved a much higher social welfare than all other benchmark approaches. Figure 2 shows the ratios of the social welfare of approaches $\mathbf{FD}$, $\mathbf{CM}$ and $\mathbf{DN}$ to that of $\mathbf{CS}$ in networks of different scales in which each MC&C controls 6 radars ($[12, 2]$ implies 12 radars with 2 MC&Cs). We can see that 1) $\mathbf{DN}$ achieved slightly lower social welfare than $\mathbf{CM}$ if there are a small number of radars (e.g., 12) and 2) $\mathbf{DN}$ achieved higher social welfare other approaches if there are more than 12 radars and the advantage increases with the scale of the network. This result is intuitive since an MC&C's strategy space increases with the number of radars. Given the real time constraint, distributed optimization with coordination may achieve better performance than centralized optimization. For the two centralized approaches, $\mathbf{CM}$ achieved a higher social welfare than $\mathbf{CS}$ since $sw(\mathbf{CM})/sw(\mathbf{CS})$ is higher than 1.

We can also see that the fully distributed approach $\mathbf{FD}$ achieved much worse performance than other approaches due to lack of coordination. One interesting observation from the experiments is that

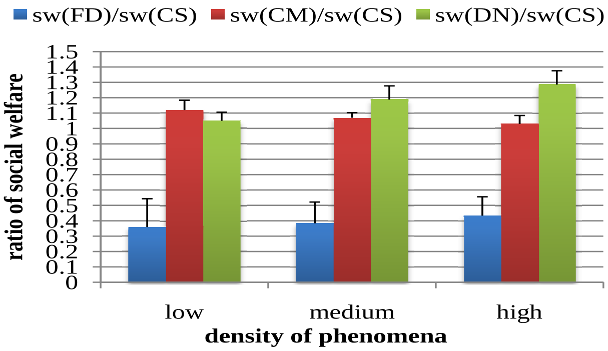

Figure 3: Average social welfare and phenomena density.

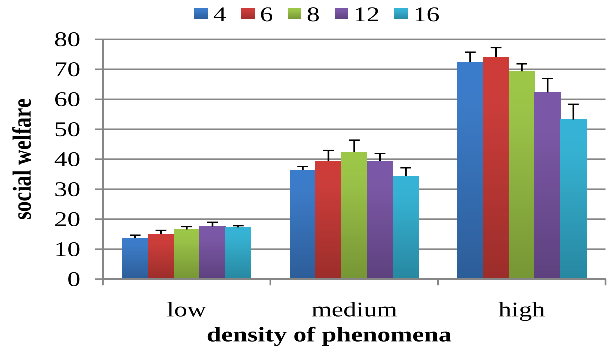

Figure 4: Average social welfare and network structure.

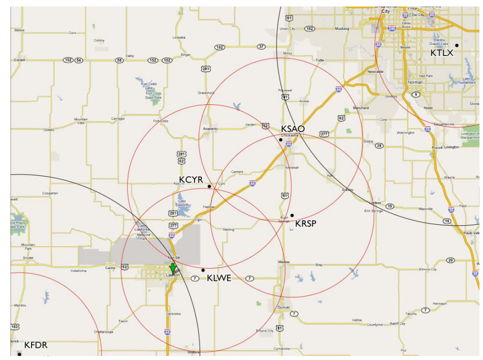

Figure 5: Location of the 4 IP1 radar nodes in Oklahoma.

$\mathbf{FD}$'s social welfare based on its *own* goal set is very high. However, $\mathbf{FD}$'s social welfare based on the *oracle* goal set is low, which is partially due to *belief propagation*: if an MC&C has wrong belief about the real phenomena, it may still have wrong (or even worse) belief after it sends out scanning commands based on its old wrong belief. Through coordination, MC&Cs will "talk" to each other and accordingly, they may have a more accurate understanding of real phenomena.

6.4.2 *Density of phenomena and network structure*

We found through simulation that the density of phenomena had a large effect on the performance of different approaches. It is intuitive since, with more phenomena, more goals will be generated and the search space of the optimization problem increases. We use the average number η of phenomena per radar at each heartbeat to measure the density of phenomena. For our domain, an η in the range of $[0.5, 1]$ (respectively, $[1, 3]$ and $[3, 6]$) is considered as low (respectively, moderate and high). It can be found from Figure 3 that the advantage of $\mathbf{DN}$ over the other approaches increases with the increase of the phenomena density. In addition, for different phenomena densities, $\mathbf{CM}$ achieved a higher social welfare than $\mathbf{CS}$, which had a much better performance than $\mathbf{FD}$.

One important objective of simulation is to investigate how the performance of $\mathbf{DN}$ is affected by the network structure, i.e., the number of radars controlled by each MC&C. Intuitively, if an MC&C has to control a large number of radars, it cannot find a good solution given its heartbeat deadline. However, if each MC&C has only a small number of radars, an MC&C can find a local optimal solution but the global solution based on all MC&Cs' local optimal solutions may be much worse than the global optimal solution. Figure 4 shows how the performance of $\mathbf{DN}$ is affected by the number of radars controlled by each MC&C in a network with 48 radars. It can be found that 1) when the phenomena density is low, it is better to allow each MC&C to control relatively more radars (e.g., 12,); 2) when the phenomena density is medium, it is better to allow each MC&C to control around 8 radars; and 3) when the phenomena density is high, it is better to allow each MC&C to control a small number radars (e.g., 4, 6).

6.4.3 *Speed of phenomena and boundary goals*

We also observed that the advantage of $\mathbf{DN}$ over the other approaches increases with the increase of the ratio of boundary goals and moving speed of phenomena (figures omitted due to space limitation). With more boundary goals, coordination between MC&Cs becomes more important since it may improve the utilities of these boundary goals by removing redundant scans and having multiple radars to observe the same phenomenon. If a phenomenon moves fast, multiple MC&Cs may need to coordinate with each other to satisfy the goal existing for multiple heartbeats.

7. FIELD STUDY

We have implemented our approach on the IP1 Testbed, which is located in southwestern Oklahoma in the heart of tornado alley. Figure 5 shows the location and coverage area of the testbed. The testbed consists of four mechanically steered parabolic dish X-band radars atop small towers. The circles around KSAO, KCYR, KLWE, and KRSP show the 30 km coverage area of the IP1 radars. The nearest NEXRAD sites located near the IP1 testbed are the radars at Twin Lakes (KTLX) and Frederick (KFDR) and are shown here with 40 km and 60 km range rings. An interested reader can refer to [5, 15] for the IP1 system architecture.

We evaluated our approach during the 2010 CASA Spring Experiment from April 1st to June 15th. This time period corresponds to a yearly maximum of severe storms and tornadic weather in our testbed domain. We reran cases archived during this experiment period from severe weather events using our system emulator which simulates the behavior of the system in a non-closed loop fashion - that is we can verify the behavior of the scan optimization, but the supplied radar data is from the canned case, not from an actual regeneration of data using, for example, a radar simulator. Using this system emulation approach we verified the scanning behavior of the goal-based multi-step optimization.

Figure 6 shows one example of the scanning pattern for each radar from an emulated test case. On the left of Figure 6 shows the scanning actions of radars using our approach and on the right of Figure 6 shows radars' scanning actions using the previous approach described in [3]. Each radar does a pie shaped sector scan, the number of arcs on the edge denoting the number of elevation angles in the scan. It can be found in Figure 6 that the two approaches very often, but not always generated different scanning commands for the radars.

To further test goal-based multi-step optimization versus a baseline functionality of the system we disabled the previous system's time-since-last-scanned scan optimization heuristic and reran test cases where we compared this baseline system versus the fully

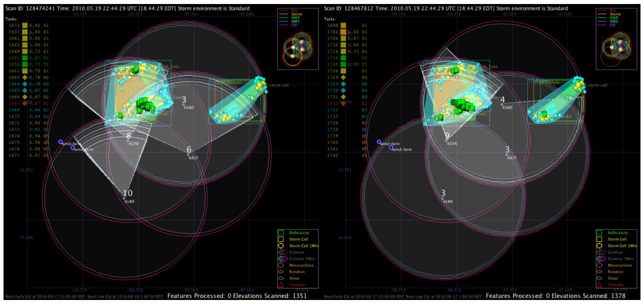

Figure 6: Scanning pattern for each radar. This snapshot was taken from data on May 19th 2010 22:44:29 (UTC).

functional goal-based multi-step optimization. We observed that the myopic baseline configuration generally exhibited a much "greedier" approach to scanning - often performing repeated scans of the same phenomena while ignoring other scan requirements such as satisfying user needs for regular low-level surveillance scans. These results with real data verified our simulation studies and showed that 1) goal based problem formulation more precisely models the needs of multiple end-users and 2) multi-step optimization together with negotiation based coordination efficiently schedules radars' scanning actions over multiple heartbeats.

It has been observed that our approach is significantly better at meeting the user specified "time-since-last-scan" requirements. In addition, our new approach avoided redundant scanning important phenomena and found phenomena failed to be observed by the old approach. Besides the numerical values that were obtained through the analysis of the real-time scanning actions generated by MC&Cs, we also got direct feedback from domain experts informing us that because we were using predictions of the future locations of phenomena we also were doing a better job of scanning the "leading edges" of storm systems. This is an important benefit because most of the interesting observables that lead to better warning by humans are located in these areas.

8. CONCLUSION

This paper presents a distributed resource allocation model combining heuristic search and asynchronous negotiation. In more detail, our contributions to the state of the art include:

- We introduce the concept "goal" to model end-users' preferences over multiple heartbeats and cast the complex sensing resource allocation problem as a continuous time optimization problem. The goal based formulation enhances modularity and improves the adaptivity of our approach to changing environments and user preferences. Each MC&C utilizes a genetic algorithm to find its local optimal strategy over multiple heartbeats given its neighbor MC&Cs' current strategies.
- We extended the distributed negotiation model in [3] by allowing MC&Cs to 1) exchange "plans" over multiple heartbeats and 2) make tradeoffs regarding local optimization time, negotiation time, and concurrency.
- We empirically show that our approach achieved better performance than some benchmark approaches.
- We have applied our approach to an operational radar testbed that is deployed in Oklahoma to observe severe weather events and it has exhibited much better performance than previous techniques.

Future research directions include improving the distributed resource allocation model. For instance, MC&Cs can make multilateral agreement through mediation that allows neighboring MC&Cs to make moves concurrently. It is also possible that the negotiation stops with a local optimal solution and it may be beneficial to accept some poor agreements to help in the long run. Our ongoing research will also focus on applying this framework to other large scale real-time optimization problems.

9. ACKNOWLEDGMENTS

We thank Michael Krainin and Chongjie Zhang for their help with building the simulator and designing the genetic algorithm. This work was supported primarily by the Engineering Research Centers Program of the National Science Foundation under NSF Cooperative Agreement No. EEC-0313747. Any Opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the National Science Foundation.

10. REFERENCES

[1] P. Cramton, Y. Shoham, and R. Steinberg, editors. *Combinatorial Auctions*. MIT Press, 2006.

[2] Y. Kim, M. Krainin, and V. Lesser. Application of max-sum algorithm to radar coordination and scheduling. In *Proc. of the Twelfth International Workshop on Distributed Constraint Reasoning*, pages 5–19, 2010.

[3] M. Krainin, B. An, and V. Lesser. An application of automated negotiation to distributed task allocation. In *Proceedings of the 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pages 138–145, Nov. 2007.

[4] V. Lesser, C. Ortiz, and M. Tambe, editors. *Distributed Sensor Networks: A Multiagent Perspective*, volume 9. Kluwer Academic Publishers, May 2003.

[5] D. McLaughlin and Coauthors. Short-wavelength technology and the potential for distributed networks of small radar systems. *Bulletin of the American Meteorological Society*, 90:1797–1817, 2009.

[6] P. J. Modi, W.-M. Shen, M. Tambe, and M. Yokoo. Adopt: asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence*, 161:149–180, 2004.

[7] R. Nair, P. Varakantham, M. Tambe, and M. Yokoo. Networked distributed POMDPs: a synthesis of distributed constraint optimization and POMDPs. In *Proceedings of the Twentieth National Conference on Artificial Intelligence*, pages 133–139, 2005.

[8] A. Rogers, D. Corkill, and N. Jennings. Agent technologies for sensor networks. *IEEE Intelligent Systems*, 24(2):13 –17, 2009.

[9] E. Ruzanski, Y. Wang, and V. Chandrasekar. Development of a real-time dynamic and adaptive nowcasting system. In *Proc. of the International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology*, 2009.

[10] T. Sandholm. An implementation of the contract net protocol based on marginal cost calculations. *Eleventh National Conference on Artificial Intelligence*, pages 256–262, January 1993.

[11] L.-K. Soh and C. Tsatsoulis. Reflective negotiating agents for real-time multisensor target tracking. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1121–1127, 2001.

[12] R. Stranders, A. Farinelli, A. Rogers, and N. R. Jennings. Decentralised coordination of mobile sensors using the max-sum algorithm. In *Proc. of the 21st international jont conference on Artifical intelligence*, pages 299–304, 2009.

[13] M. Yokoo and K. Hirayama. Distributed Breakout Algorithm for Solving Distributed Constraint Satisfaction Problems. In *Proceedings of the Second International Conference on Multiagent Systems*, pages 401–408, 1996.

[14] W. Zhang, G. Wang, Z. Xing, and L. Wittenburg. Distributed stochastic search and distributed breakout: properties, comparison and applications to constraint optimization problems in sensor networks. *Artificial Intelligence*, 161(1-2):55–87, 2005.

[15] M. Zink, E. Lyons, D. Westbrook, J. Kurose, and D. Pepyne. Closed-loop architecture for distributed collaborative adaptive sensing of the atmosphere: Meteorological command & control. *International Journal for Sensor Networks*, 6(4), 2009.