Coordination in Disaster Management and Response: a Unified Approach

Myriam Abramson, William Chao, Joseph Macker, Ranjeev Mittu

Naval Research Laboratory, Washington DC 20375, USA
myriam.abramson@nrl.navy.mil

Abstract. Natural, technological and man-made disasters are typically followed by chaos that results from an inadequate overall response. Three separate levels of coordination are addressed in the mitigation and preparedness phase of disaster management where environmental conditions are slowly changing: (1) communication and transportation infrastructure, (2) monitoring and assessment tools, (3) collaborative tools and services for information sharing. However, the nature of emergencies is to be unpredictable. Toward that end, a fourth level of coordination – distributed resource/role allocation algorithms of first responders, mobile workers, aid supplies and victims – addresses the dynamic environmental conditions of the response phase during an emergency. A tiered P2P system architecture could combine those different levels of coordination to address the changing needs of disaster management. We describe in this paper the architecture of a tiered P2P agent-based coordination decision support system for disaster management and response and the applicable coordination algorithms including a novel, self-organized algorithm for team formation.

1 Introduction

Large scale disasters are characterized by catastrophic destruction of infrastructure (e.g., transportation, supply chain, environmental, communication, etc). The lack of coordination characterizes such disasters. While preparedness is the best response to emergencies [1], a multiagent-based approach to coordination decision support systems (CDSS) can play an important role in disaster management and response (DM&R) in shaping decentralized decision-making on a large scale. However, the diverse aspects of coordination make it difficult to find a unified approach for continuous control. Coordination is at best defined as an emergent property from local interactions, either cooperative or competitive, explicit or implicit, in the pursuit of multiple goals. A taxonomy of coordination is illustrated in Fig. 1. Finding a unified approach is a key problem in disaster management because a cooperative approach in the preparedness phase has to be complemented with a competitive approach in the response phase due to life-threatening situations requiring fast and reactive solutions. For example, satellite-based environmental surveillance requires centralized planning and scheduling in advance due to geo-spatial and atmospheric constraints but needs to be supplemented by unmanned aerial vehicles for timely information requests. Conversely, planning and preparedness decisions have to be relevant in emergency situations to the first responders and provide them with guidelines. A disaster management task is specified by the tuple $\{P, T, A, S\}$ where P is the set of plans, T the set of tasks or incidents, A the set of agents, volunteers, first responders, and coordinators, S the set of sensors, static or mobile, and where $A \subseteq S$. The problem consists of matching the needs of T with the resources of A in a decentralized and concurrent fashion to accomplish goals defined by P.

This paper is organized as follows. First, we explain the agent-based CDSS framework in Sect. 2 and motivate a tiered peer-to-peer (P2P) coordination architecture for integrating the different coordination dimensions of DM&R. Then, in Sect. 3, we introduce two basic coordination algorithms for heterogeneous agents suitable in disaster management response. In Sect. 4, a self-organized, semi-centralized coordination algorithm is introduced in support of the architecture proposed. An empirical evaluation follows in Sect. 5 on a canonical fire/rescue scenario to illlustrate the relative merits of the coordination algorithms. Finally, Sections 6 and 7 conclude with related work.

2 Agent-Based CDSS

Recent technological advances in communication and processing power, enabling sensor networks and personal digital assistants, have made possible the self-organization of mobile agents (robots or people) and

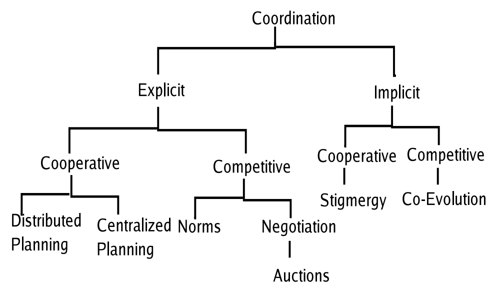

Fig. 1. Coordination Taxonomy and related coordination mechanisms

geo-localized decision support. The complexity of decentralized decision-making is tamed by delegating certain management tasks to proxy agents [2]. Coordination is a pervasive management task that helps reduce interference in role assignments and enhance information sharing. The degree of consensus to obtain before making a decision can be arbitrarily set. The lower the degree of consensus, the more flexible the agents are in reacting to outside events and making timely decisions but the more negative interactions can occur. Assuming rational communicating and trusting agents reduces the degree of consensus overhead required in coordination tasks because the agents are likely to reach the same conclusions given the same information.

Current collaborative web-based tools have essentially a fixed client/server approach because of the relatively stable nature of internet routing. Coordination is achieved through the server as a synchronizing blackboard passively mediating the interactions of intelligent agents as clients. P2P approaches, such as JXTA [3], de-emphasize the role of the server as passive synchronizer but the role of mediator is taken up actively by "rendez-vous" peers and "relays." Peers discover each other through advertisements propagated by relays and rendez-vous peers. This suggests a flexible, semi-centralized coordination architecture for complex tasks such as DM&R where the preparedness and information sharing architecture can seamlessly adapt to rapidly changing conditions and communication infrastructure (Fig. 2). In this framework, coordination at the network layer, whereby a host is chosen to act as relay for propagating messages through the network, maps with a coordinator role at the application layer.

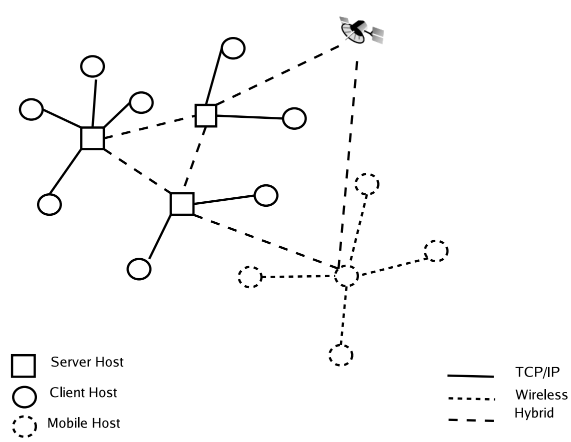

Fig. 2. Semi-Centralized Coordination Architecture

3 Coordination Algorithms for Heterogeneous Agents

One of the key coordination problem in disaster management is the heterogeneity of the players involved. Roles provide a convenient a priori decomposition of a task and are a key coordination tool [4]. Roles can be

Algorithm 1 Basic agent loop for cooperative distributed systems

```
set initial role to explore
active ← true
rounds ← 0
while (no termination condition) do
if (active) then
  sense environment
  act according to role
  broadcast information to neighbors
  active ← false
else
  read neighbors' new information, if any
  deliberate and select role
  active ← true
endif
rounds++
end while
```

viewed either as fixed slots in a team structure that are filled by agents or part of an agent's behavior repertoire in its relationships with other agents that can determine the structure of a team. The decision complexity for the role allocation of N agents to p tasks is $O(p^T)$ where T is the number of teams of size t that can be selected from N agents:

$$T = \begin{cases} \binom{N}{t} & case\,1: \; homogeneous\,agents \\ \binom{N+t+1}{t} & case\,2: \; heterogeneous\,agents \end{cases} \tag{1}$$

t is the number of roles in a team which might not correspond to the number of agents N. In the first case, $\sum_i^T t_i \leq N$, agents have distinct, mutually-exclusive roles. In the second case, agents fill a number of non mutually-exclusive roles (i.e. an agent can perform a number of roles in a team). The complexity in role allocation scales up with heterogeneous agents where the mapping of agents to roles is one-to-many. The basic agent loop for role allocation in distributed cooperative systems is described in Alg. 1. Two basic matching algorithms for generalized role allocation of heterogeneous agents running in polynomial time are described below.

3.1 Greedy Set Cover Algorithm

This is an approximate matching algorithm [5] that finds the minimum set cover for a list of resources needed to accomplish a task given the initial capabilities of a set of agents sorted in maximal task coverage order (Fig. 3). In addition, a small penalty is given to capabilities not relevant to the task. The preference for a task is proportional to its coverage and the preference of the agents selected for the task, ensuring commitment to mostly completed tasks.

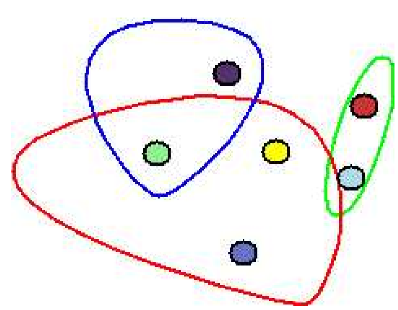

Fig. 3. Greedy Set Cover of a task decomposed into 6 needs (dots) with 3 agents and one overlapping capability.

3.2 Hungarian Algorithm

The "Hungarian" algorithm for weighted bipartite matching [6,7] solves constraint optimization problems such as the job assignment problem in polynomial time. An implementation of this algorithm follows Munkres' assignment algorithm [8]. The algorithm consists of transforming a weighted adjacency matrix of $roles \times agents$ into equivalent matrices until the solution can be read off as independent elements of an auxiliary matrix. While additional rows and columns with maximum value can be added to square the matrix, the optimality is no longer guaranteed if the problem is over-constrained, i.e. there are more roles to be filled than agents. This algorithm can be extended to heterogeneous agents by expanding the original set of agents to virtual homogeneous agents, one for each capability required by the task, ignoring other capabilities. The mapping of agent capabilities to incident needs is illustrated by the bipartite matching graph in Fig. 4. Here too, the preference for a task depends on the coverage of the task and the preference of the agents selected for the task.

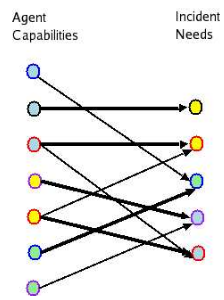

Fig. 4. Weighted Bipartite Matching

4 Semi-Centralized Coordination Algorithms

Semi-centralized algorithms were found to be both practical and efficient in the large-scale coordination of agents [9] and lend themselves well to a contract net protocol. The level of specificity in the planning of large groups do not extend to individual behaviors. DM&R planning in the National Response Plan [10] provides specific guidelines at the lowest geographical and organizational level but leaves room for self-organization. Semi-centralization through team formation enables the continuous control of decentralized decision-making to achieve planned objectives and maintain situation awareness through data fusion at the global level.

Clustering techniques are often used in team formation to find appropriate subteams minimizing intra-goal distance and maximizing inter-goal distance. Adaptive clustering techniques extend fixed clustering techniques such as k-means to dynamic conditions when the clusters change over time. The problem is (1) to self-adjust to the correct number of clusters, i.e. the proper degree of centralization, and (2) incrementally update the clusters. We first describe a cycle-based self-organizing algorithm for the formation of "cluster heads" and then our extension of this algorithm to take into account environmental demands.

4.1 Low-Energy Adaptive Clustering Hierarchy (LEACH) Algorithm

The LEACH algorithm [11] is a stochastic adaptive algorithm for energy-efficient communication in wireless sensor networks in the task allocation of aggregating and routing messages back to the base station (BS). Because of the limitation on battery power, the task should be fairly distributed among the nodes. In addition, aggregating the data to reduce noise before sending it to the BS is more efficient. Rotating this "cluster head" role among the nodes will (1) minimize the overall energy consumed and (2) allow the battery power to get replenished through solar energy. A round in the algorithm includes a setup phase establishing a transmission schedule to maximize bandwidth utilization and a steady-state phase where data fusion occurs and the aggregated messages are actually transmitted. It is assumed that the percentage of nodes that should take up this role is known a priori by the agents. An algorithm where the agents take turn assuming the "cluster head"

Algorithm 2 LEACH algorithm

```
active←true
rounds←0
set activation rate
while (no termination condition)
if (active) then
  if (cluster_head) then
    read BS mesgs
    aggregate BS mesgs
    send BS mesgs to BS
    rounds←0
    cluster_head←false
  else
    route BS msgs to elected cluster_head
 endif
  generate a random number r
  T←estimate threshold
  if (r < T) then
    cluster_head←true
    broadcast advertisement messages
  endif
  propagate messages
  active←false
else
  if (cluster_head) then
    read election mesgs
    create transmission schedule
    broadcast transmission schedule
  else
    read advertisement mesgs
    elect cluster_head
    send election mesg to cluster_head
  endif
  active←true
endif
rounds++
end while
```

role is described in Alg. 2. An agent i assumes the role of "cluster head" if the stochastic probability is below a threshold T, determined as follows:

$$T(i) = \begin{cases} \frac{P}{1-P*(r\,mod\frac{1}{P})} & if\, n \in G \\ 0 & otherwise \end{cases} \tag{2}$$

where P is the desired percentage of cluster heads known a priori, r is the current round, and G is the set of agents that have not been cluster heads for the past $\frac{1}{P}$ rounds. If below threshold, the agent will advertise its services. Otherwise, the agents elect as their leader the closest agent according to the advertisements received.

4.2 Extension of the LEACH Algorithm

This algorithm assumes that (1) the activation percentage is given a priori and (2) the activation duration during which a schedule is propagated and messages are transmitted to the base station is fixed. This works well for sensor networks (e.g. unmanned aerial vehicles) where the number of nodes is known in advance and the only mission is to report back to the BS. This algorithm needs to be adapted to act as a relay in a mobile ad hoc network (i.e, transmit messages from any node to any other nodes) and to autonomously adjust to the number of nodes in the network. The time interval allocated to be a "cluster head" need not be limited to a single transmission to the BS and has to adapt to the needs of the network. First, we motivate our approach with an illustration of self-organization by the El Farol Bar problem and then describe the Adaptive Team Formation (ATF) algorithm.

El Farol Bar [12] The problem addressed was that of a paradox: how agents could coordinate without communication by not going to the same place, the El Farol bar, at the same time. This problem addresses “tragedy of the common” type of problems where resources, otherwise plentiful, become scarce without implicit coordination. Arthur evolved a set of hypotheses relating previous attendance at the El Farol bar and the “happiness” outcome based on how crowded the bar was found to be. Note that an agent has complete information when going to the bar. The online co-adaptation of those hypotheses to recent situations rather than convergence was the key to achieving coordination. Other work in this area found that coordination could be achieved with less variance if the agents relied on the accuracy of the same adaptive gradient algorithm[13].

Adaptive Team Formation (ATF) algorithm If an agent i does not assume a network role or “cluster head,” it will receive only advertisement messages, and will send only election messages. As long as it receives advertisement messages, it does not have to compete for the network role. However, if everybody assume the same strategy, no service will be provided. The key idea is to predict the correct individual phase to alternate between roles based not only on internal disposition but also on the state of the environment. A coverage metric as the number of agents reached over the total number of agents looking for the service measures the performance of this algorithm. The time-to-live (TTL) parameter, latency and communication range affect the propagation of messages and the coverage of a node.

In contrast to other adaptation problems where convergence of an agent to a fixed behavior (or role) is desired, congestion problems like the El Farol Bar problem requires learning when to change behavior to resolve conflicts. Response thresholds in swarm intelligence [14] induce a dynamic task allocation depending (1) on the disposition of the agents and (2) the environmental demands. A simple reinforcement learning scheme allocates agents to the task by either raising or lowering their response threshold. In our problem, an increase in connectivity (due to proximity or communication range) should sensitize an agent to be a team leader but a decrease in advertisement messages should also be an incentive to assume the role. The stimulus s_i for an agent i to become a team leader at time t depends on the connectivity of the agents (i.e. the number of other agents within one hop) or degree d_i of the network node, the change in connectivity δ_i, and repulsion factor $\alpha \in (0,1)$ as follows:

$$s_{i_0} = \frac{d_{i_0}}{d_{i_0}+1} \tag{3}$$

$$s_{i_{t+1}} = s_{i_t} - \alpha \frac{\#Advertisements}{\#Elections + \#Advertisements + \epsilon} + \delta_{i_{t+1}} \tag{4}$$

Here $\epsilon > 0$ is a small constant that prevents division by 0. The agent’s response threshold T_i at time t taking into account its internal disposition θ_i and external demands is then as follows:

$$T_{i_t} = \frac{\theta_{i_t}}{1+e^{-s_{i_t}}} \tag{5}$$

The initial disposition $\theta_{i_o} \in (0,1)$ of an agent can be a function of its battery power or other hardware capabilities. To avoid specialization and redistribute the manager task fairly among the agents according to their capabilities, θ_t is adjusted based on the “fatigue” of performing the task or the “boredom” of not performing the task measured in cycles as in the LEACH algorithm above (see Subsect. 4.1).

$$\theta_{i_t} = \theta_{i_0} * (r \, mod \, \frac{1}{\theta_{i_0}}) \tag{6}$$

where r is the number of elapsed rounds.

It is assumed that the agents can perform their deliberative task in selecting a role in one round and that roles are noncommittal. The leader determines a proper role allocation (Sect. 3) of the team by iterating through

Algorithm 3 Adaptive Team Formation

```
active ←true
set repulsion rate
rounds ← 0
while (no termination condition) do
if (active) then
  read role allocation mesg
  perform step(s) in role
  generate a random number r
  update stimulus and disposition
  T←estimate threshold
  if (r < T) then
    leader←true
    broadcast advertisement mesg
  else
    leader←false
    elect leader
    send election mesg to leader
  endif
  send role preferences mesgs to leader
  propagate all messages
  active←false
else
  read role preferences mesgs
  read role allocation mesgs
  read election mesgs
  read advertisement mesgs
  if (leader) then
    optimize task allocation for team
    send role allocation mesgs
  endif
  active←true
endif
rounds++
end while
```

each task. Roles are allocated to the best ranking team based on coverage of the task and preferences. The process repeats again on the remaining agents and tasks until no team can be formed. Redundancy against message loss occurs when roles are reallocated either by the same manager agent in the next round or another manager agent. Algorithm 3 describes the combined process.

5 Experimental Evaluation

The experiments were conducted with RePast [15], an agent-based simulation and modeling tool where agents act concurrently in a decentralized manner on a $n \times n$ grid. Its powerful scheduling mechanism was used to model the asynchronous behavior of the agents. Communication between agents was implemented by transmitting messages to agents in a Moore neighborhood[1] of 7 cells, eliminating cycles, and time-to-live parameter set to 6 hops. In addition, a 5% message loss proportional to distance was simulated.

Figure 5 compares the comparative coverage rates of the LEACH and ATF clustering algorithms without task allocation for a varying number of agents in fixed random locations on a 100×100 grid. Only cluster nodes relay messages to other agents. The agents were randomly initialized with a disposition rate varying in the [0,0.1] range. The swarm-based ATF algorithm provides a significantly better coverage albeit with a larger clustering rate for each node. Nodes were cluster nodes at a rate of ~0.5% in the LEACH algorithm while their rate was evaluated at ~0.8% in the ATF algorithm.

[1] All agents within a specified square radius in all directions are included.

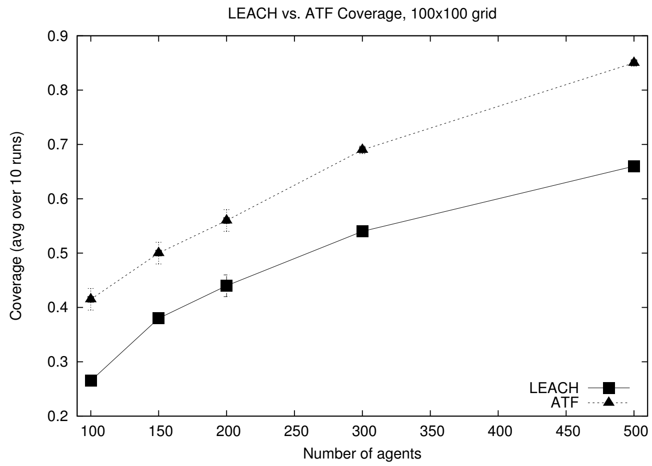

Fig. 5. LEACH vs. ATF coverage comparison over 100 cycles

5.1 Coordination Metric

Because coordination is an emergent property of interactive systems, it can only be measured indirectly through the performance of the agents in accomplishing a task where a task is decomposed in subgoals. The more complex the task, the higher the number of subgoals needed to be achieved. While performance is ultimately defined in domain-dependent terms, there are some common characteristics. Performance in a task can be measured either as the number of steps taken to reach the goal, i.e. its time complexity, or as the amount of resources required, i.e. its space complexity. An alternative evaluation for coordination is the absence of "failures", for example negative interactions such as collisions or lost messages. Figure 6 illustrates the taxonomy of coordination solution quality in pursuit games. To show the scalability of a solution, the evaluation must linearly increase with the complexity of the task [16].

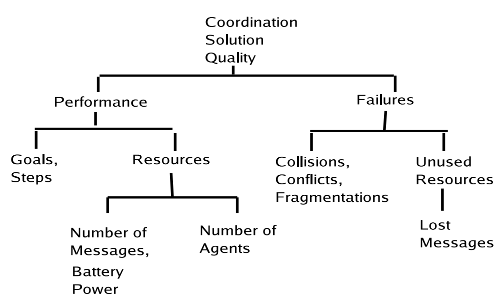

Fig. 6. Taxonomy of coordination solution quality

A combined coordination quality measure is defined as the harmonic mean of goals achieved g, resources expanded r and collisions c as follows :

$$g = \frac{\#Goals\ Achieved}{\#Goals} \tag{7}$$

$$r = \frac{\#agents}{log_2(\#Messages\ Received + 1) + \#agents} \tag{8}$$

$$c = \frac{\#agents}{log_2(\#Collisions + 1) + \#agents} \tag{9}$$

$$coordination = \frac{3grc}{gr + rc + cg} \tag{10}$$

Such a metric combining the different aspects of coordination can evaluate the tradeoff of performance and consuming bandwidth in large-scale tasks. In [17], coordination is evaluated solely as an effort, such as additional steps to avoid collisions or messages to avoid role conflicts, and do not take into account the indirect effect on performance.

5.2 Fire/Rescue Problem

In our scenario, buildings are randomly created on a $n \times n$ grid with a random probability of being on fire and of spreading fire to adjacent buildings if not extinguished in time. Each fire or incident creates an emergency situation requiring up to k types of resources. In turn, each responder agent can provide up to k matching types of capabilities. There are a total of n capabilities and needs for each agent and incident ($n < k$). The problem consists of dynamically matching capabilities and needs with a team of agents. When a team of agents with the desired capabilities is situated near the incident within the agent's perception range p, the emergency will be removed. There are no scheduling constraints in matching resources but the overall resource requirements might increase over time as the fire spreads. Each agent has a perception range p and a typically greater communication range h to communicate with its neighbors. Figure 7 is an illustration of the simulation of this domain in Repast.

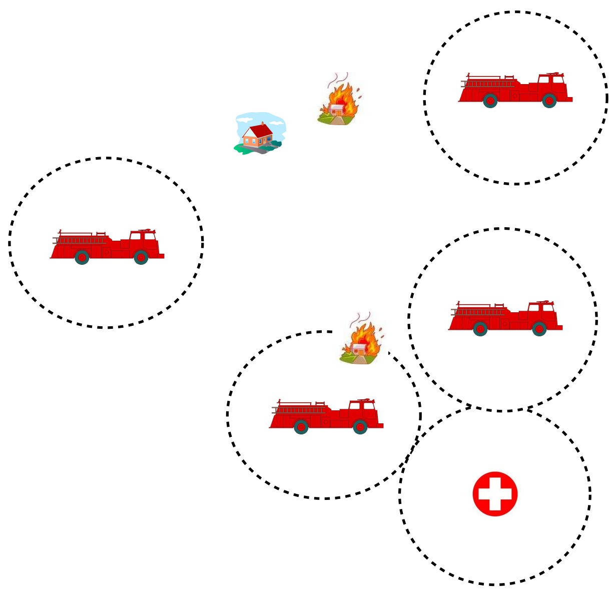

Fig. 7. Fire/Rescue Scenario

There are 4 types of messages in this scenario. Advertisement messages are broadcast while election messages are point-to-point. Received messages not matching the destination host are ignored using flooding, while ATF retransmits the message to the next leader node. Role preferences are communicated among agents (point-to-point to the leader with ATF and broadcast otherwise) that include the known targets and the associated preferences for covering each resource needed. The preferences are based on the distance to the incident and reflect the expected utility of the agent's capabilities. When observing an incident, a "resource needed" message is propagated among the agents describing the incident.

Figure 8 shows the coordination performance (10) of the two decentralized role allocation algorithms (Sect. 3), Greedy Set Cover and "Hungarian," in this domain along with a random strategy of just stumbling upon an incident while exploring. A simple flooding algorithm was used to transmit messages among agents. A significant difference was found between those two algorithms (t-test p-value = 0) and the random strategy. The results suggests that approximate algorithms using appropriate heuristics (Greedy Set Cover) perform better in complex and uncertain environments than non-optimal solutions to non-approximate algorithms ("Hungarian").

Figure 9 shows the coordination performance (10) in this scenario with ATF where the elected leader node performs the network role of relaying messages. In the semi-centralized case, the leader node performs the managerial task of role allocation. In addition, one scout agent transmits additional observations to the leader nodes with retransmission to clients. In the distributed case, the role allocation task is performed implicitly by the agents. Both cases use the Greedy Set Cover algorithm. Results show that self-organization and semi-centralization of role allocation incurs an overhead with a large number of agents and depends on other information available to the leader node for its performance with a smaller number of agents. There is a significance difference under ATF with 50 agents (t-test p-value = 0.002) between fully distributed role allocation and semi-centralized role allocation using the Greedy Set Cover algorithm.

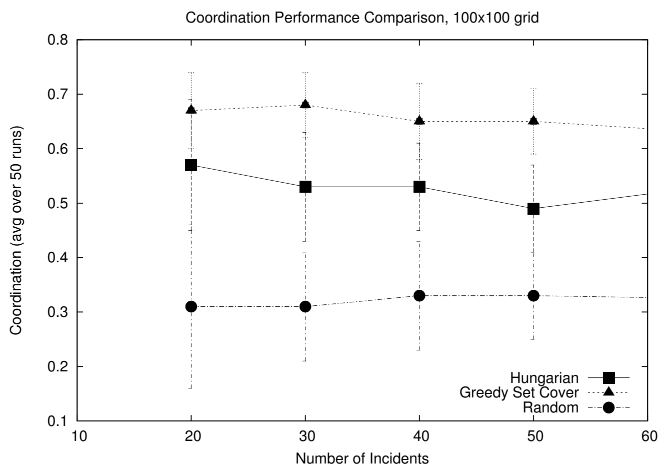

Fig. 8. Comparative coordination performance of decentralized role allocation of 100 agents and varying task complexity from 20 to 50 incidents on a 100x100 grid ($n \leq 4$, k=8, cycles=100) with flooding.

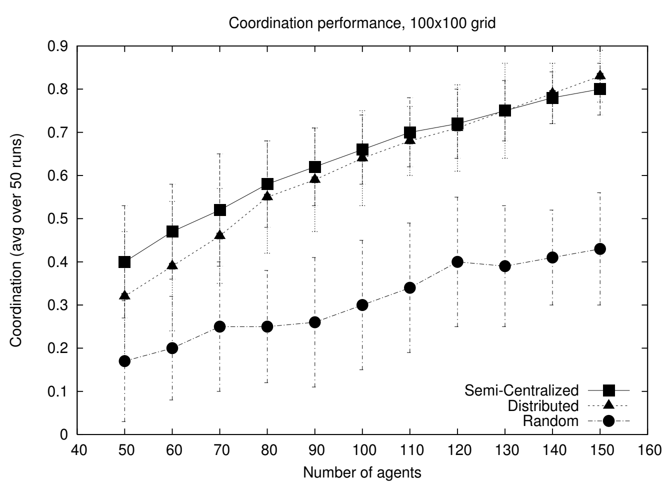

Fig. 9. Comparative coordination performance with ATF ($\alpha = 0.05$) for 20 incidents and a varying number of agents. Greedy Set Cover was used for the semi-centralized and distributed case.

6 Related Work

Workpad [18] has proposed a 2-layer P2P architecture where the static internet backend provides the information services necessary to first responders in a tethered mobile ad hoc network. The scenarios explored an architecture for a coordination layer on top of the network layer where a team leader would reallocate tasks to solve predicted fragmentation of the network due to the mobility of the agents. In this paper we explored in detail the algorithms for role allocation and for selecting a team leader in a self-organized way.
Cooperative mediation [19] combines distributed role allocation with partial centralization. An agent, acting as mediator, recommends value changes (role assignments) to neighboring agents to optimize local subproblems. If a current solution is different from an optimal solution, the mediator transmits repairs to the relevant agents. Agents are prioritized to act as mediator based on the size of their "social knowledge." If a solution cannot be found, the neighboring agents transmit their constraints which could involve other agents enlarging the context of the subproblem. Cooperative mediation achieves a global optimal solution in a distributed way by exploiting the substructure of the problem. If no local optimal solution can be found, the mediator will progressively enlarge its context until an optimal global solution is found. Similarly, the ATF approach uses the degree of connectivity as a stimulus to influence the tendency of an agent to be a team leader but the election of a leader is explicit. A team leader divides the search space according to the substructure of the problem but does not attempt to reach a more global solution in this paper. The role of the team leader is not only to coordinate other agents in solving a task but also to coordinate the information sharing between agents.

7 Conclusion

We have presented applicable coordination algorithms and introduced a tiered P2P architecture to unify the different communication and coordination dimensions of DM&R with possible applications to other complex environments such as battlespace management. In addition, a novel self-organized, semi-centralized algorithm, ATF, has been introduced extending the LEACH algorithm to adaptive team formation. Semi-centralization is important to achieve planned objectives with bounded resources and to integrate disparate systems. Experimental evaluations of role allocation algorithms for heterogeneous agents have been presented in the fire/rescue domain along with a coordination metric that takes into account communication costs as well as partial goals achieved. Dynamic coordination alternating between semi-centralized and distributed role allocation based on this coordination metric will be examined. Future work should include a more complex scenario where the leader nodes communicate between themselves to reach a more global solution.

References

1. The National Commission on Terrorist Attacks: The 9/11 Commission Report. http://www.9-11commission.gov (2004) last accessed, 02/01/2007.
2. Scerri, P., Pynadath, D., Schurr, N., Farinelli, A., Gandhe, S., Tambe, M.: Team oriented programming and proxy agents: The next generation. Workshop on Programming MultiAgent Systems, AAMAS 2003 (2003)
3. Project JXTA. http://www.jxta.org last accessed, 02/01/2007.
4. Abramson, M.: Three myths about roles. Technical report, American Association of Artificial Intelligence, Arlington, VA (2005) Fall Symposium Workshop on Roles, an Interdisciplinary Perspective: Ontologies, Programming Languages, and Multiagent Systems.
5. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. 2nd edn. MIT Press (2001)
6. Kuhn, H.W.: The hungarian method for the assignment problem. Naval Research Logistics Quarterly **2**(83) (1955)
7. Papadimitriou, C.H., Steiglitz, K.: Combinatorial Optimization: Algorithms and Complexity. Dover Publications (1998)
8. Burgeios, F., Lassalle, J.C.: An extension of the munkres algorithm for the assignment problem to rectangular matrices. Communication of the ACM **14** (1971) 802–806
9. Scerri, P., Vincent, R., Mailler, R.: Comparing three approaches to large-scale coordination. In: Proceedings of the Large-Scale Coordination Workshop at AAMAS. (2004)
10. Department of Homeland Security: National Response Plan. http://dhs.gov/xprepresp/publications last accessed, 01/23/2007.

11. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: Proceedings of the 33rd Hawaii IEEE International Conference on System Sciences. (2000)
12. Arthur, W.B.: Inductive reasoning and bounded rationality. American Economic Review (1994)
13. Bell, A.M., Sethares, W.A.: Avoiding global congestion using decentralized adaptive agents. IEEE Transactions on Signal Processing **49**(11) (2001)
14. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: from Natural to Artificial Systems. Oxford University Press (1999)
15. North, M.J., Collier, N.T., Vos, J.R.: Experiences creating three implementations of the repast agent modeling toolkit. ACM Transactions on Modeling and Computer Simulation **16**(1) (January 2006) 1–25
16. Durfee, E.H.: Scaling up agent coordination strategies. IEEE Computer (2001)
17. Rosenfeld, A., Kaminka, G.A., Kraus, S.: Adaptive robotic communication using coordination costs for improved trajectory planning. In: Proceedings of the American Association of Artificial Intelligence. (2006)
18. Mecella, M., Catarci, T., Angelaccio, M., Buttarazzi, B., Krek, A., Dustdar, S., Vetere, G.: Workpad: an adaptive peer-to-peer software infrastructure for supporting collaborative work of human operators in emergency/disaster scenarios. In: IEEE International Symposium on Collaborative Technologies and Systems (CTS 2006). (2006)
19. Mailler, R., Lesser, V.: Solving distributed constraint optimization problems using cooperative mediation. In: International Conference on Autonomous Agents and Multiagent Systems. (2004)