Challenges in Distributed Coalition Formation among Collaborative Multi-Agent Systems:

An Experimental Case Study on Small-World Networks

Predrag T. Tošić and **Naveen K. R. Ginne**
Department of Computer Science, University of Houston, Houston, Texas, USA
ptosic@uh.edu, ginne.naveen@gmail.com

Abstract - *We study collaborative multi-agent coalition formation in a fully decentralized, resource-bounded setting. In that context, we analyze, simulate and optimize* Maximal Clique based Distributed Coalition Formation (MCDCF) *algorithm. We continue our experimental, simulation-based investigation of MCDCF performance on types of underlying graphs that various research communities find important. In particular, we focus in this paper on MCDCF performance on the* small-world graphs. *We consider three classes of such graphs, where each class is defined by the probability of randomly rewired (that is, non-local) edges. We compare and contrast performances of the MCDCF protocol as the fraction of randomly re-wired edges is increased from zero (corresponding to k-regular rings) to 20% of the total number of edges in the graph. We discuss those aspects of coalition formation performance that turned out in accordance with our expectations, as well as those where the performance was not as expected. The main conclusions are twofold. One, the exact structure of sparse-on-average graphs can make a great deal of difference when it comes to the scalability and efficiency of our graph partitioning based approach to coalition formation. Concretely, the algorithm performs much better on Erdos-Renyi random graphs with modest average node degrees and several hundreds of nodes than on small-world graphs of the same or comparable densities and sizes. Two, some of the candidate coalition search mechanisms in MCDCF need to be improved in order to ensure better overall performance when our algorithm is run on small-world-like topologies. We hope to report improvements in that regard in our near-future work.*

Keywords: distributed algorithms, distributed AI, multi-agent systems, coalition formation, small-world networks

1 Introduction and motivation

We study, devise and analyze scalable distributed graph algorithms that are useful in addressing challenging and important coordination problems in *Distributed Artificial Intelligence* (DAI). We are interested in artificial agents such as softbots, robots, unmanned vehicles or smart sensors that are fully autonomous, capable of communicating with each other, and therefore potentially able to self-organize in various ways in order to accomplish tasks that would exceed the computational, sensing and/or acting capabilities of the individual agents. Of our particular interest are mechanisms and protocols that enable large ensembles of such autonomous agents to coordinate with each other in a *fully decentralized* manner, and collaborate in order to accomplish various tasks. The kinds of agents we have in mind are referred to as *distributed problem solvers* (or DPS agents) in the DAI literature [2, 22, 23].

Among various interesting problems in distributed coordination and control of such ensembles of DPS agents, we have been extensively studying genuinely autonomous, fully distributed, dynamic multi-agent *coalition formation* [15 – 18, 20, 21]. We have approached coalition formation from two perspectives: one is the *multi-agent coordination* view of coalition formation, and the other is the *distributed graph algorithm design* view. We have integrated these two views and designed an original, fully distributed, scalable and robust coalition formation graph algorithm appropriately (if somewhat awkwardly) named, *Maximal Clique based Distributed Coalition Formation* (MCDCF) [15]. In this paper, we build on the top of our earlier work on MCDCF [15-17, 19-21], summarize and interpret our recent simulation results on small-world underlying networks, and outline our ongoing and near-future research on further improvements and scaling-up of our algorithm.

The rest of this paper is organized as follows. In Section 2, we motivate the problem of multi-agent coalition formation, briefly review some approaches to this problem found in the literature, and outline the problem setting that we have been studying in our prior and ongoing work. In Section 3, we summarize our MCDCF algorithm for coalition formation among collaborative agents. In Section 4, we present, analyze and discuss our recent simulation results when MCDCF is run on three types of small-world graphs. Section 5 summarizes the paper and indicates some of the major challenges ahead.

2 Distributed coalition formation

Distributed coalition formation in multi-agent domains is an important coordination and collaboration problem that has been extensively studied by the Multi-Agent Systems (MAS) research community (e.g., [1, 2, 8-10, 12, 15-21]). There are many important collaborative MAS applications where autonomous agents need to form groups, teams or coalitions. Reasons behind multi-agent coalition formation can vary. In the context of *Distributed Problem Solving* (DPS) collaborative agents, possible motivations include, but are not limited to, the following: to share resources, to jointly

complete tasks that exceed the abilities of individual agents, and/or to improve some system-wide performance metric such as the speed of joint task completion [7, 12, 17].

One well-studied general coalition formation domain is a collaborative multi-agent environment populated with a variety of distinct mutually independent tasks, where each task requires a tuple of resources on the agents' part in order for the agents to be able to complete that task [10-12, 17, 18]. In this *distributed task allocation* context, agents need to form coalitions such that each coalition has sufficient cumulative resources or capabilities across the coalition members in order to be able to complete the assigned task.

While *distributed problem solving* and, in particular, *distributed task or resource allocation* are perhaps the most common MAS contexts in which coalition formation has been studied, the problem of designing and analyzing protocols for efficient coalition formation among autonomous agents has arisen in other contexts, as well. In the most general setting, the conceptual and mathematical foundations of the coalition formation process have been addressed from the standpoint of *algorithmic game theory* and *mechanism design* [22, 23]. Game-theoretic models and techniques have been considered particularly valuable when the agents that engage in forming coalitions belong to different designers and/or organizations; such agents in general tend to be *self-interested* (as opposed to *altruistic*).

Among other reasons why agents may engage in cooperative coalition formation, limitations on sensing, computational (processing and/or memory), energy (e.g., batter life in wireless sensors; or fuel tank capacity in UAVs), communication and acting resources of individual agents have been prominent in much of the prior research on various types of multi-agent encounters [8, 23]. A thorough survey of the state of the art (covering most of the important directions of coalition formation research up to approximately year 2005) can be found in [7].

We study the problem of distributed coalition formation in the following problem setting. We assume a collaborative multi-agent, multi-task dynamic and partially observable environment. The tasks are assumed *mutually independent* of each other. In general, different tasks may have different values or utilities associated with them; moreover, the utility of a particular task may be differently perceived by different agents. This problem setting is particularly appropriate for many MAS applications involving *team robotics* and *autonomous unmanned vehicles* [5, 17, 18]. In particular, our agents are distributed problem solvers: they are assumed to be *strictly collaborative*, not selfish. The agents have certain capabilities that (may) enable them to service various tasks. Similarly, the tasks have certain resource or capability requirements, so that no agent or coalition of agents whose joint capabilities do not meet a particular task's resource requirements can serve that task [10, 16, 17]. Each task is of a certain *value* to an agent. Agents are assumed capable of communicating, negotiating and making agreements with each other [2, 8, 10, 17, 22]. Communication is accomplished via exchanging messages. This communication, however, is not free, which is why it is desirable to limit the amount of communication per agent [17]. An important difference between our problem setting and that in [10] is that we assume that *an agent's resources are not transferable to other agents* [15-17]. Thus, the only way for an agent A_i to use the internal resources of agent A_j for the purpose of servicing some task is that A_i and A_j join the same coalition, and then jointly attack that task.

Our *distributed maximal clique-based coalition formation* algorithm is based on the idea that, in a *peer-to-peer* MAS, an agent would prefer to form a coalition with those agents that it can communicate with directly, and, moreover, where every member of such potential coalition can communicate with any other member *directly*. That is, the preferable coalitions are (maximal) cliques [15-17]. Finding a maximal clique in an arbitrary graph is **NP**-hard in the centralized setting [3, 4]. This implies the computational hardness that, in general, each agent faces when trying to determine the maximal clique(s) it belongs to. However, if the degree of a node is sufficiently small, then finding some or even all maximal cliques this node belongs to may be feasible. If one cannot guarantee that all the nodes in a given underlying network topology are of small degrees, then the system designer and/or deployment coordinator may want to impose additional constraints, in order to ensure that the agents' coalition formation according to the MCDCF protocol is actually computationally feasible [15-17]. Alternatively, the system designer may want to consider modifying the objective function, so that the candidate coalitions an agent would consider forming need not necessarily be *maximal cliques* that this agent belongs to. The most appropriate design choice clearly depends on the application at hand, the nature of agents' tasks, and the agents' communication and computational resources [17].

3 The MCDCF algorithm

The MCDCF coalition formation protocol [15-18] is a *distributed graph algorithm*. The abbreviation stands for *Maximal Clique based Distributed Coalition Formation*, and was originally introduced in [15]. The underlying undirected graph captures the communication network topology among the agents. Each agent is a node in the graph. The necessary requirement for an edge between two nodes to exist is that the two nodes be able to directly communicate with one another. (Note: while communication links in practice are directional, the MCDCF algorithm works for both directed and undirected underlying graphs.) Communication takes place by local or *group broadcasts*. The group broadcast nature of the assumed communication model is primarily inspired by the applications that drove the original MCDCF design, namely team robotics and autonomous unmanned vehicles, in particular *micro-UAVs* [18, 19].

The basic idea behind MCDCF is to *partition* the underlying graph into (preferably, maximal) *cliques* of nodes. These maximal cliques would in practice usually also need to satisfy some additional criteria in order to form coalitions of desired quality. These coalitions are then maintained until they are no longer preferred by the agents - that is, when they are no longer sufficiently useful or even meaningful.

To ensure overall scalability and computational cost feasibility of the protocol, the candidate coalitions in MCDCF are required to be cliques of uniformly bounded sizes. This requirement, in practice, means one of the two possibilities. One possibility is that it is *a priori* known that the underlying MAS network topology is such that it can be guaranteed that there are no cliques that are prohibitively large. Otherwise, the system designer, based on the application at hand and the available system resources, a priori chooses a threshold $K = K(n)$, where n is the total number of nodes, such that only the coalitions of sizes up to K are considered.

Agents form coalitions in a fully distributed manner as follows. Each agent (i) first learns of who are its neighbors, then (ii) determines the appropriate candidate coalitions, that the agent hopes are (preferably maximal, but certainly of sizes bounded by K) cliques that it belongs to, then (iii) evaluates the utility value of each such candidate coalition, measured in terms of the joint resources of all the potential coalition members, then (iv) chooses the most desirable candidate coalition, and, finally, (v) sends this choice to all its neighbors. An agent also receives similar coalition proposals from some subset of its one-hop neighbors. The agent compares its current coalition proposal against those received from the neighbors. If there is an agreement among all agents in the current proposal of our agent, let's call it A, then this condition is recognized and a new coalition is formed. Once this happens, A's other neighbors, namely, those not in the newly formed coalition (if there are any such neighbors left), are notified of the formation of the new coalition. If there is no agreement, meaning, if at least one agent in the current proposal of A does not agree with that proposal, then the process continues and the protocol moves to the next round.

The basic procedure outlined above is repeated, together with all agents updating their knowledge of (a) what are the preferred coalitions of their neighbors, and (b) what coalitions have already been formed – until eventually every agent has joined some coalition. Note that, when an agent has no available coalition partners left, it necessarily forms the trivial, "singleton coalition"; hence, it is guaranteed that each agent will join some coalition after finitely many rounds (and it may possibly end up being the trivial coalition).

Once an agent has joined a coalition, that agent is done with the MCDCF execution. Once every agent has joined some coalition, the entire coalition formation process terminates. The round in which the last agent has joined some coalition is the final round; the total number of rounds until termination is our main criterion of our algorithm's computational efficiency and feasibility. We have shown in details elsewhere that, assuming a sufficiently sparse underlying network topology and under some additional, relatively mild assumptions that typically hold in application domains of our main interest, MCDCF is always guaranteed to converge after finitely many rounds [15-18].

A detailed description and pseudo-code for the algorithm can be found in [17]. Recent improvements and optimizations are summarized in [20]. These optimizations have been tested on structure-free random graphs and have generally produced very encouraging results [21]. One of the goals of the present study is to determine, just how effective these optimizations are when MCDCF is applied to *small-world-like* underlying topologies.

MCDCF is characterized by (i) being *fully decentralized* and genuinely *peer-to-peer* (P2P) and (ii) requiring only a very *coarse-level synchrony*. Property (i) is immediate: there is no central control of any kind, and no agent has any special role. In fact, all agents execute the same code. Moreover, MCDCF is a *highly local algorithm*, in a sense that there is no *network flooding* (or any other similar method of global information sharing). Due to property (i), and unlike many other multi-agent coalition formation protocols and other coordination techniques found in the MAS literature, MCDCF is rather robust and has no single point of failure. In particular, it works as intended in the presence of possible multiple node and/or communication link failures. However, these node and link failures have to be of a "nice" enough kind ; more details about the robustness and fault tolerance properties of MCDCF protocol can be found in [17].

By *coarse synchrony* we mean that no agent begins the next round before all its neighbors are done with the previous round. In particular, we assume that each agent uses a timer so that, once no response from a particular neighbor is received within the allocated time slot, that particular neighbor is removed from the agent's extended neighborhood list and, therefore, also from the current (and all future) candidate coalitions for that agent. However, within a given round, different agents may be executing different stages of that round; again, we refer the reader to [15–17] for more details.

4 MCDCF on small-world graphs

We simulate the MCDF algorithm using Java. We have conducted experiments to investigate how well MCDCF performs on small-world graphs, and also to verify general theoretical predictions related to scalability [16, 17]. The two most important parameters for the MCDF algorithm are the number of nodes n and the average node degree avg_deg. To experimentally validate (or challenge) the performance predictions that we made in our prior work based on graph-theoretic considerations [15-17], we considered two basic scenarios. In the first scenario, we maintained the number of nodes as a constant and varied the average node degree. In the second scenario, the average node degree is constant and we varied the total number of nodes.

The assumptions we made throughout the simulation experiments are as follows:

1. We have used the Watts-Strogatz model [25] to generate graphs with *Small World* structures. The graphs have been generated using the software *NetworkX*, a Python package to generate complex networks [24].
2. We have generated the small-world graphs starting with a regular k-ring and then, with probabilities 0, 0.1, and 0.2, respectively, we would rewire randomly selected local edges (in the original regular ring) and replace those selected edges with random (hence, in general, non-local) edges.

3. To make it an apple-to-apple comparison with our experimentation with Erdos-Renyi random graphs [21], we used the edge replacement operation as described in remark 2 so that we control the exact total number of edges in the graph, and hence the exact average node degree.
4. The number of rounds until convergence and the total number of coalitions in the final coalition structure are both computed as (arithmetic) means over 10 simulation runs where a different graph is generated randomly at the beginning of each run.
5. The maximum coalition size (see Figure 2 below) is the maximum size among all coalitions found across all of those 10 simulation runs (for a given pair of parameter values (*n, avg_deg*).

Scenario 1:

The varying parameter in this scenario is the average node degree and the number of nodes is held constant (n = 200). We have incremented the average node degree in multiples of two, from *avg_deg* = 4 up to *avg_deg* = 20. We summarize the main lessons learned from the simulation results for this scenario and their statistical analysis. In each plot in this subsection, the x-axis captures the average node degree of the underlying graph. Specifically, the average node degree plotted along the x-axis ranges from 4 to 20 in the increments of 2.

The plot capturing the dependence of the number of rounds on the average node degree shows a rapid growth rate when the probability of random rewiring is greater than zero (Figure 1). The MCDF algorithm has been argued in our earlier work to scale well for sparse underlying graphs, and our simulation results for Erdos-Renyi random graphs with increasing densities validate our prior theoretical predictions in [16,17]; see [21]. However, for small-world graphs with non-zero probability of randomly rewired edges, the number of rounds, at least in the range of graph densities captured in our simulations, appears to grow exponentially. This is rather alarming; one implication is that better mechanisms for how MCDCF traverses the lattice of candidate coalitions of a given node (see [20] for details) need to be sought.

We recall that, as the average node degree increases, so does the average size of a node's neighborhood list, and hence the number of candidate coalitions to consider. Moreover, the number of candidate coalitions grows *exponentially* with the number of neighbors. The mechanisms for optimizing this search of the lattice of candidate coalitions that each agent (node) locally performs are discussed in detail in [20]. While these mechanisms and optimizations have resulted in very good scaling and convergence behavior for the Erdos-Renyi graphs [21], the results for small-world graphs with non-zero probabilities of random re-wiring indicate poor scalability. That is, for the small-world graphs, the current mechanisms in MCDCF that are supposed to ensure fairly fast convergence in practice (as long as the underlying graphs are relatively sparse) may be inadequate.

For the random rewiring probabilities of 0.1 and 0.2 the convergence is much slower compared to when the rewiring probability is 0. The reason behind this is the graph generated using the Watts-Strogatz model with probability 0 is a k-regular ring with n nodes, where each node is connected to its 2k nearest neighbors [24]. This property of near-by nodes having multiple common neighbors, according to our intuition, should enable the agents to (i) find and agree to non-trivial, relatively sizable maximal cliques, and (ii) be able to do so relatively fast. These predictions have been partly validated, as can be seen from Figures 1 and 3. We also found the following interesting property of MCDCF's behavior on regular k-rings: when the probability of random rewiring is 0, we typically get three different kinds of coalitions: one relatively large coalition of the size k + 1 (where k is the number of nearest neighbors on each side of a given node, for the total of 2k), one doubleton, and the rest of the nodes end up as singleton (i.e., trivial) coalitions.

As we increase the probability of rewiring in the Watts-Strogatz model the likelihood of nodes having multiple common neighbors, in general, decreases; consequently, we expected to see both the number of rounds to form coalitions to increase, and the coalition structure quality (cf. measured by average coalition size at the termination of the algorithm) to deteriorate. These predictions have been validated; however, the slow-down in terms of what appears as an exponential increase in the number of rounds until convergence is quite alarming, as further discussed below.

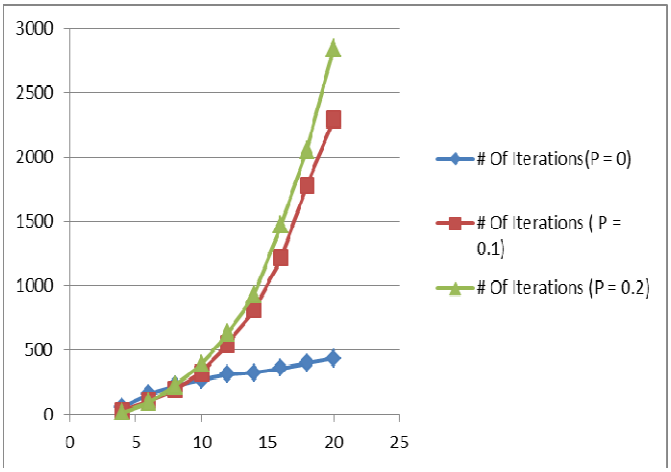

Figure 1. *Rates of convergence for three values of probability of random re-wiring in small world graphs, as a function of graph density; the y-axis is the number of rounds until convergence.*

The plot in Figure 2 captures the dependence between the average node degree and the maximum coalition size. As the density of the graph increases, the number of candidate coalitions per node also increases. We have observed in simulations that the nodes initially have many candidate coalitions that are bigger in size as the graph density is increased. However, the coalitions the agents have proposed in the earlier rounds need not be agreed by all the neighbors to whom those coalition proposals were sent. As the negotiation on coalition proposals continues and the agents fail to reach agreements round after round, those agents that have failed to reach an agreement eventually get to drop these relatively larger candidate coalitions that have been proposed but not

agreed upon; the agents then end up having to select candidate coalitions that are smaller in size. (That is, each agent's search for a candidate coalition is monotonic across the lattice of possible coalitions with respect to the usual subset relation; see [17, 20] for details). Therefore, higher average node degrees need not imply that larger-sized coalitions will necessarily be formed. However, we admittedly still fall short of complete understanding of the experimentally observed non-monotonicity in the maximum coalition sizes as the function of the average node degree (Figure 2); further investigating this phenomenon is subject of our ongoing work.

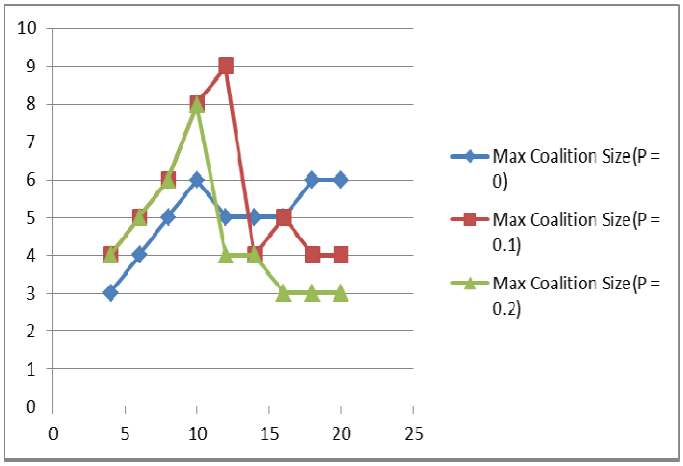

Figure 2. *Maximum coalition size as a function of graph density. The y-axis is the maximum size of an obtained coalition among all coalitions across all the runs with the same parameters values.*

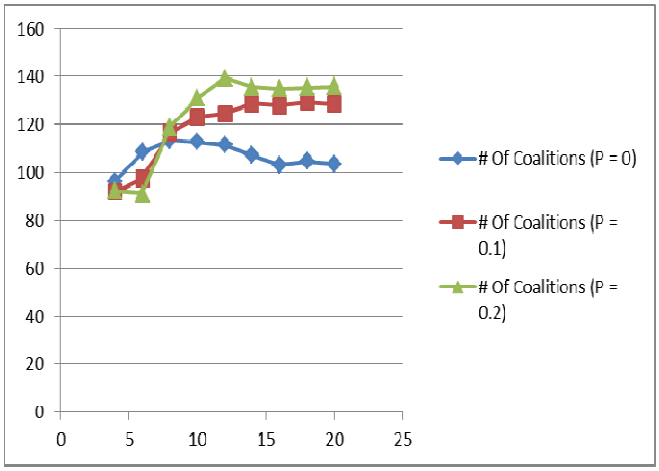

Figure 3*. The number of formed coalitions as a function of graph density. The y-axis is the average total number of coalitions formed across ten graphs with the specified parameter values.*

The plot showing the dependence of the total number of coalitions formed as a function of the average node degree also shows non-monotonic behavior (Figure 3). We observe that, even with higher average node degrees, a vast majority of nodes end up in either trivial singleton coalitions or else doubletons. In particular, coalitions of size 3 or greater are few and far between. We therefore find the quality of the resulting coalition structures, in general, to be below our expectations for this class of underlying graphs.

This behavior may be an indication of potentially serious issues with how MCDCF handles small-world graphs: as discussed earlier, its mechanisms for optimizing search of the lattice of candidate coalitions and the tie-breaking mechanisms (when a node has two or more *equally preferable* candidate coalitions to choose from) appear to work very well for the *structure-free* random (Erdos-Renyi) graphs [21], but the MCDCF performance on non-trivial small-world graphs of the same sizes and densities as those in [21] is much worse. Therefore, we need to further investigate new optimization mechanisms for MCDCF that would enable our protocol to better handle the small-world graphs.

Scenario 2:

In this scenario we vary the number of nodes and the average node degree remains constant: *avg_node* = 10 throughout. We increment the number of nodes by 20, starting from n = 40 up to n = 200 nodes. As in *Scenario 1*, for each pair of values (n, *avg_deg*), we run MCDCF on ten different, randomly generated graphs. Just like in *Scenario 1*, the captured values for the metrics of interest in Figures 4 – 5 are the actual arithmetic averages across all ten runs for a given pair of parameter values n and *avg_deg*. We do not show the plot for the maximum obtained coalition size as a function of the total number of nodes (i.e., the analog of Figure 2 in the previous subsection), as we did not find any systematic dependence or correlation between those two problem parameters. We next share several important observations and what we consider to be the main lessons learned from this scenario.

In Figure 4, we see that the number of rounds to reach convergence increases approximately *linearly* with the total number of nodes. According to theoretical predictions [16, 17], in general, how long it takes for MCDCF to converge primarily depends on the neighborhood sizes, not the total number of nodes in the graph. More specifically, how long will a typical node take to agree with some subset of its neighbors on which coalition to form, is cf. determined by the *neighborhood size* and hence (i) the size of the initial candidate coalition and (ii) the total number of candidate coalitions that may be considered during the protocol's execution. However, we recall that the total number of rounds is, in essence, determined by the slowest agent to form a coalition. Therefore, it is not surprising that as the number of agents increases, the convergence slows down (at a sensible rate) even though the average neighborhood sizes are held fixed: more nodes overall likely in practice results in more "slow" nodes, hence slower convergence. This increase in the average number of rounds with an increase in n is only slightly more pronounced when some random edges are added. In particular, even for regular k-rings, the convergence slows down linearly with an increase in the total number of nodes. As the probability of random edge re-wiring is increased, the linear slow-down with increase in n is still observed, but with slightly greater slopes. While not

necessarily surprising, we certainly find this experimental insight interesting.

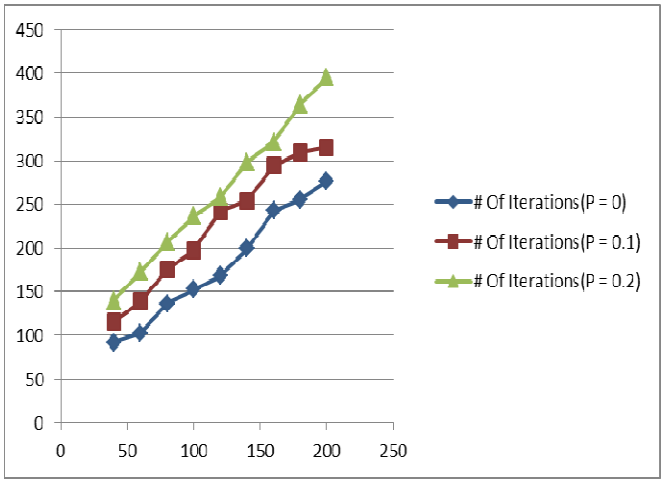

Figure 4. *Convergence rates for the three studied values of probability of random rewiring as a function of the total number of nodes. The y-axis is the average number of rounds until convergence.*

The plot in Figure 5 captures the relationship between the number of nodes and the total number of coalitions formed; the number of coalitions is a *monotonically increasing* function of *n* as well as of the probability of rewiring, as we expected. Moreover, the dependence on the number of nodes appears to be linear. This is in general accordance with our predictions, since the *fixed* average node degree implies that we shouldn't expect more larger-sized coalitions as *n* increases; rather, we get more coalitions of roughly the same average sizes as for smaller *n*. Therefore, this is one aspect of MCDCF performance where the theoretical predictions, our conceptual intuition and our subsequent simulation experimentation are all in agreement.

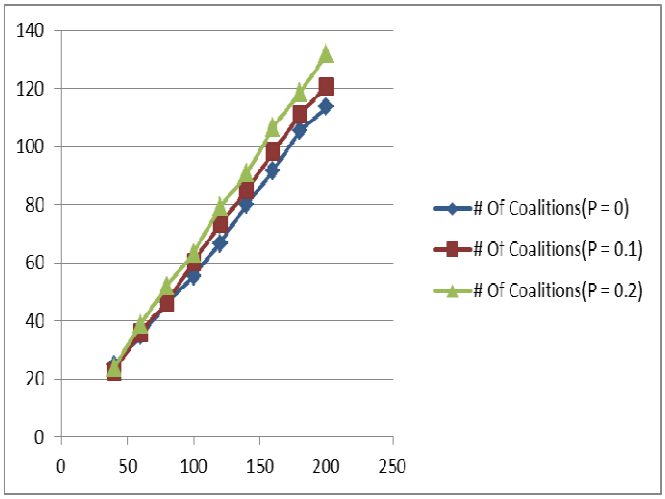

Figure 5: *Number of coalitions formed as a function of the total number of nodes. The y-axis is the average number of coalitions formed for the randomly generated small-world graphs with the specified number of nodes.*

We also note that, for rewiring probabilities of 0.1 and 0.2, due to random nature of our test graphs, most nodes end up in singleton and doubleton coalitions basically irrespectively of *n* and *avg_deg* (see also Figure 3). Therefore, in particular, as the number of nodes and the probability of rewiring increase, the total number of formed coalitions in general also tends to increase.

We recall that, in the original MCDCF algorithm, a node selects the *lexicographically first* coalition among the two or more equal candidate coalitions, as the tie breaking mechanism [16, 17]. This method works fine when the underlying graph does not have any particular structure – for instance, for random (Erdos-Renyi) graphs [21]. In the case of the Watts-Strogatz small-world graph model [24], when the probability of rewiring is 0, then the graph is a regular *k-ring* (*k* neighbors to the left, *k* neighbors to the right) with *n* nodes, where every node forms edges with its *k* nearest neighbors on each side. As we increase the probability of rewiring, the graph becomes more "random", but there's still structure to it – namely, the small world structure; in theory, this small-world property holds as long as the probability of rewiring is strictly less than 1 [25].

Lexicographically selecting one among several equally good candidate coalitions, in this case where the graph has a small-world structure, apparently adversely affects the performance of MCDCF. Hence, instead of using the Lex ordering as the tie breaking mechanism as we did in our prior work incl. our experiments with random graphs, we have recently started experimenting with selecting a new candidate coalition *uniformly at random* among the equally preferable candidate coalitions. Our early results indicate that random selection of the next candidate coalition tends to result in fewer singleton coalitions and therefore better resulting coalition structure overall. We however leave further discussion until we obtain stronger experimental evidence of this tentative improvement.

5 Summary and future work

We have applied our distributed coalition formation algorithm to network topologies that, to various extents, exhibit the small-world structure. We have tested our algorithm on randomly generated small-world graphs with up to 200 nodes. Our simulations have produced several generally expected results, which to a considerable (but not full) extent validate prior claims that our approach to coalition formation via a particular way of distributed graph partitioning is applicable to a broad variety of underlying graphs [15, 17, 20].

However, we have also obtained some rather unexpected and, in some instances, disappointing results. One lesson learned is that the optimizations of MCDCF reported in [20], while very helpful when the algorithm is run on certain types of graphs such as structure-free random graphs [21], appear inadequate when MCDCF is run on the small-world graphs. Therefore, one immediate objective of our ongoing research is to improve inter-agent coalition negotiation as well as individual agent's candidate coalition search mechanisms, in

order to ensure faster convergence, as well as better resulting coalition structures, when our algorithm is run on small-world-like underlying networks. We also need to investigate in more depth what properties of small-world graphs cause subpar performance of MCDCF with respect to some of the metrics discussed in this paper.

Plans for the future work, in addition to further analyzing and fully understanding some unexpected aspects of the MCDCF performance on the small-world graphs, include studying our algorithm's performance on the *power-law graphs* of comparable sizes and densities to the random graphs we studied in [21] and small-world graphs studied in the present paper. Additionally, we are very interested in experimentally investigating robustness and fault-tolerance properties of our algorithm in the presence of individual node and/or link failures. In that context, it is of a particular interest to understand how the graph structure (for example, Erdos-Renyi vs. small-world vs. power-law graphs) affects robustness of MCDCF. We hope to report new insights on these interesting questions in the near future.

References

[1] S. Abdallah, V. Lesser. *Organization-Based Cooperative Coalition Formation, in* Proc. IEEE / WIC / ACM Int'l Conf. on Intelligent Agent Technology (2004)

[2] N. M. Avouris, L. Gasser (eds.). *Distributed Artificial Intelligence: Theory and Praxis*, Euro Courses Comp. & Info. Sci. vol. 5, Kluwer Academic Publ. (1992)

[3] T. H. Cormen, C. E. Leiserson, R. L. Rivest. *Introduction to Algorithms*, MIT Press (1990)

[4] M. R. Garey, D. S. Johnson. *Computers and Intractability: a Guide to the Theory of NP-completeness*, W.H. Freedman & Co., New York (1979)

[5] M. Jang, S. Reddy, P. Tosic, L. Chen, G. Agha. *An Actor-based Simulation for Studying UAV Coordination*, in 16th European Simulation Symposium (ESS '03), pp. 593-601, Delft, The Netherlands (2003)

[6] X. Li, L.K. Soh. *Investigating reinforcement learning in multiagent coalition formation*, TR WS-04-06, AAAI Workshop Forming and Maintaining Coalitions & Teams in Adaptive MAS (2004)

[7] D. de Oliveira. *Towards Joint Learning in Multiagent Systems Through Opportunistic Coordination*, PhD Thesis, Univ. Federal Do Rio Grande Do Sul, Brazil (2007)

[8] T.W. Sandholm, V.R. Lesser. *Coalitions among computationally bounded agents*, in *Artificial Intelligence* vol. 94, pp. 99-137 (1997)

[9] T. W. Sandholm, K. Larson, M. Andersson, O. Shehory, F. Tohme. *Coalition structure generation with worst case guarantees*, in *AI Journal* vol.111 (1-2), pp. 209-238 (1999)

[10] O. Shehory, S. Kraus. *Task allocation via coalition formation among autonomous agents*, in Proceedings of IJCAI-95, Montréal, Canada, pp. 655–661 (1995)

[11] O. Shehory, S. Kraus. *Methods for task allocation via agent coalition formation,* in *AI Journal* vol. **101** (1998)

[12] O. Shehory, K. Sycara, S. Jha, *Multi-agent coordination through coalition formation*, in *Intelligent Agents IV: Agent Theories, Architectures & Languages*, LNAI #1365, pp.153-164, Springer (1997)

[13] L. K. Soh, X. Li. *An integrated multilevel learning approach to multiagent coalition formation*, in Proc. Int'l Joint Conf. on Artificial Intelligence IJCAI'03 (2003)

[14] R. Sun. *Meta-Learning Processes in Multi-Agent Systems*, in *Intelligent agent technology: research and development*, N. Zhong, J. Liu (eds.), pp. 210-219, World Scientific, Hong Kong (2001)

[15] P. Tosic, G. Agha. *Maximal Clique Based Distributed Group Formation Algorithm for Autonomous Agent Coalitions*, in Proc. Workshop on Coalitions & Teams, AAMAS-2004, New York City, New York (2004)

[16] Tosic, P., Agha, G. *Maximal Clique Based Distributed Coalition Formation for Task Allocation in Large-Scale Multi-Agent Systems*, Post-Proceedings of MMAS-04; LNAI vol. 3446, pp. 104-120, Springer (2005)

[17] P. Tosic. *Distributed Coalition Formation for Collaborative Multi-Agent Systems,* MS thesis, Univ. of Illinois at Urbana-Champaign (UIUC), Illinois, USA (2006)

[18] P. Tosic, G. Agha. *Understanding and Modelling Agent Autonomy in Dynamic Multi-Agent, Multi-Task Environments*, in Proc. EUMAS '03, Oxford, England (2003)

[19] P. Tosic et al., *Modeling a System of UAVs on a Mission,* session on intelligent agent systems & technologies, in Proc. 7th World Multiconference on Systemics, Cybernetics & Informatics SCI'03 (2003)

[20] P. Tosic, N. Ginne. "MCDCF: A Fully Distributed Algorithm for Coalition Formation in Collaborative Multi-Agent Systems", Proc. WorldComp ICAI'10, Las Vegas, Nevada, USA (2010)

[21] P. Tosic, N. Ginne. "A Scalable Distributed Graph Algorithm for Dynamic Coalition Formation: A Performance Case Study on Random Graphs". Technical Report (2011)

[22] G. Weiss (ed.) *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence,* MIT Press (1999)

[23] M. Wooldridge. *An Introduction to Multi-Agent Systems*, Wiley (2002)

[24] The Watts-Strogatz small-world graph generator that we used can be found at http://networkx.lanl.gov/reference/generated/networkx.generators.random_graphs.watts_strogatz_graph.html

[25] Duncan J. Watts. "The "New" Science of Networks", *Annual Review of Sociology*, vol. 30, pp. 243-270 (2004)