Distributed Protocols for Multi-Agent Coalition Formation: A Negotiation Perspective

Predrag T. Tošić and Carlos Ordonez

University of Houston, 4800 Calhoun Road, Houston, Texas, USA

Abstract. We investigate collaborative multi-agent systems and how they can use simple, scalable negotiation protocols to coordinate in a fully decentralized manner. Specifically, we study multi-agent *distributed coalition formation*. We summarize our past and ongoing research on collaborative coalition formation and describe our original distributed coalition formation algorithm. The present paper focuses on negotiation-based view of coalition formation in collaborative *Multi-Agent Systems* (MAS). While negotiation protocols have been extensively studied in the context of competitive, self-interested agents, we argue that negotiation-based approach may be potentially very useful in the context of collaborative agents, as well – as long as those agents, due to limitations of their sensing and communication abilities, have different views of and preferences over the states of the world. In particular, we show that our coalition formation algorithm has several important, highly desirable properties when viewed as a negotiation protocol.
Keywords: *collaborative multi-agent systems, multi-agent coordination, negotiation protocols, coalition formation, distributed consensus*

1 Introduction and Motivation

We study autonomous artificial agents such as softbots, robots, unmanned vehicles and smart sensors, that are fully autonomous and capable of interacting and communicating with each other, and capable of self-organizing in order to accomplish tasks that may exceed the abilities of the individual agents. We are particularly interested in mechanisms that enable relatively large ensembles of such autonomous agents to coordinate with each other in a fully decentralized manner, and collaborate in order to jointly complete various tasks. This kind of agents are often referred to as *distributed problem solvers* (DPS) in the Distributed AI literature [2, 25, 26]. Coalition formation in DPS multi-agent domains is an important coordination and collaboration problem that has been extensively studied by the Multi-Agent Systems (MAS) community [1, 2, 11–14]. There are many important collaborative MAS applications where autonomous agents need to form groups, teams or coalitions. The agents may need to form coalitions in order to share resources, jointly complete tasks that exceed the abilities of individual agents, or improve some system-wide performance metric such as the speed of task completion [9, 15].

Among various interesting problems in distributed coordination and control of such agent ensembles, we have been extensively studying the problem of genuinely autonomous, dynamic and fully decentralized coalition formation [19–23]. So far, we have approached coalition formation from two perspectives common in the MAS literature: one is the conceptual multi-agent coordination view of coalition formation, and the other is the practical distributed graph algorithm design. We have integrated these two aspects and proposed a novel, fully distributed, scalable coalition formation graph algorithm named *Maximal Clique based Distributed Coalition Formation* (MCDCF for short) [19, 20]. In the present paper, we build on the top of our earlier work on designing, analyzing, simulating and optimizing the MCDCF algorithm, but this time we approach the coalition formation in general, and the MCDCF coalition formation protocol in particular, from a fresh, negotiation-focused perspective.

The rest of this paper is organized as follows. In Section 2, we summarize our MCDCF algorithm for coalition formation among collaborative agents, emphasizing the graph algorithmic and coordination aspects of that algorithm. In Section 3, we first motivate the negotiation view of coalition formation, and make the case for usefulness of such approach even when applied to DPS agents that are *altruistic* "by definition" (as opposed to *self-interested*), as long as those agents are *locally constrained* in terms of their perceptions of and actions in the environment [18, 21]. We then cast MCDCF in negotiation terms, and show that it satisfies several highly desirable properties expected from good general-purpose multi-agent negotiation protocols. Section 4 summarizes our main contributions and outlines the future work.

2 The MCDCF Protocol and Its Main Properties

We address the problem of distributed coalition formation in the following setting. We assume a multi-agent, multi-task dynamic and partially observable environment. The tasks are assumed mutually independent of each other. 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 applications involving team robotics and autonomous unmanned vehicles; see, e.g., [5, 20, 24]. Our agents are assumed to be strictly collaborative, not selfish. The agents have certain capabilities that (may) enable them to service the 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 [13, 20, 21]. Each task is of a certain value to an agent. Agents are assumed capable of communicating, negotiating and making agreements with each other [2, 11, 14]. Communication is accomplished via exchanging messages. This communication is not free: an agent has to spend time and effort in order to send and receive messages [21]. Our distributed maximal clique based coalition formation algorithm is centered at the idea that, in a *peer-to-peer* (in particular, *leaderless*) 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 [19]. That is, the preferable coalitions are (maximal) cliques. However, finding a maximal clique in an arbitrary graph is **NP**-hard in the centralized setting [4]. This implies the computational hardness that, in general, each node in the graph faces when trying to determine the maximal clique(s) it belongs to. However, if the degree of a node is sufficiently small, then finding all maximal cliques this node belongs to can be expected to be feasible. If one *a priori* does not know if all the nodes in a given MAS interconnection topology are of small degrees, then one may need to impose additional constraints in order to ensure that the agents are not attempting to solve an intractable problem [20].

The MCDCF algorithm originally introduced in [19, 20] is a distributed graph algorithm. The underlying graph captures the communication network topology among the agents, as follows. 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. The group broadcast nature of the adopted communication model is primarily due to the application domains that drove the original MCDCF design, namely team robotics and autonomous unmanned vehicles, in particular micro-UAVs [18, 21]. The basic idea behind MCDCF is to efficiently and in a fully decentralized manner partition this graph into (preferably, maximal) cliques of nodes. These maximal cliques would usually also need to satisfy some additional criteria in order to form temporary 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. In an actual MAS application, the MCDCF algorithm may need to be invoked a number of times as a coordination subroutine [19, 20].

Agents form coalitions 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) 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. This basic procedure is then repeated, together with all agents updating their knowledge of (a) what are the preferred coalitions of their neighbors, and (b) what coalitions have been already formed. In a nutshell, MCDCF is a distributed graph algorithm that defines a negotiation protocol among collaborative, but locally constrained, autonomous agents. Each agent maintains some simple data structures capturing what it knows about other, near-by agents. These data structures include neighborhood lists, candidate coalitions, the current coalition choice (i.e., the current proposal of a coalition), and some auxiliary flags via which the most important aspects of the previous rounds of the protocol are summarized. As in any negotiation protocol, what an agent proposes to some nonempty subset of other agents, may or may not get accepted by other agents. In particular, MCDCF can be also viewed as a protocol that solves a variant of *distributed consensus* problem [6, 25]: a coalition proposal becomes an actual, agreed upon coalition only if all members of the proposed coalition agree on that coalition, and moreover they do so at the same time – that is, during the same round of the MCDCF execution. More details on distributed algorithmic aspects on how MCDCF goes through its rounds until each agent has joined some coalition can be found in our prior work [19, 20, 22, 23]. The *negotiation view* of MCDCF is further elaborated upon in the rest of the present paper.

We have rigorously established elsewhere that MCDCF is guaranteed to converge in a finite number of rounds, where an upper bound on that number of rounds can be determined as a function of the total number of agents, N, and the maximum neighborhood size in the underlying communication topology graph [20, 21]. When it comes to defining *optimality* of a coalition structure, we have adopted "the bigger, the better" principle – but always under the constraint that each coalition has to be a clique, i.e., that all of its members need to be within a single hop from each other. More details on the graph-theoretic assumptions and the constraint optimization aspects of MCDCF can be found in [19–22].

3 The Negotiation View of Coalition Formation

Most of the existing literature on multi-agent negotiation assumes *self-interested agents* [6, 25, 26]. Such agents may still want, in certain situations, to cooperate with each other in various ways, including but not limited to forming teams or coalitions with each other. However, the motivation behind such cooperation is still selfish: so, in the context of coalition formation among self-interested agents, an agent will join a coalition with one or more other agents only if joining that coalition increases the agent's expected individual utility (or some other measure of that agent's individual payoff) [20, 25]. Consequently, when it comes to negotiation protocols for coalition formation among self-interested agents, the protocol needs to specify how are the "spoils" of cooperation going to be distributed among the agents. Algorithmic game theory has provided the theoretical framework for systematically addressing these complex issues [6, 25].

We observe, however, that a need for negotiation may arise among non-competing agents, as well – provided that those collaborative, non-selfish agents see the world differently. In particular, if different agents have different local views of the relevant aspects of their environments (such as tasks, resources, other agents, etc.), then it can be expected that these different agents will have different preferences on how would they like the world to be, and what should be the next step (action or sequence of actions) to bring about the most desirable state of the world [26]. These remarks would hold even in purely collaborative, *distributed problem solving* (DPS) settings where agents share a global objective or global utility function to be optimized, and do not have individual utilities per se [18, 20]. In other words, the need to negotiate might be due simply to the fact that the agents see the world differently, not necessarily that they are self-interested in the usual sense from economics or game theory. This is of considerable practical importance, since in many collaborative MAS applications involving the real-world software, robotic or other types of autonomous agents, locally bounded sensing and communicating abilities, and hence "world-views", are a rule rather than an exception [21]. In such *locally constrained* MAS settings, it is to be expected that different agents will perceive the world differently, and hence have different preferences over the possible states of the world. Thus, negotiation may be a useful methodology to reach *distributed consensus* among strictly collaborative agents. Examples of distributed applications where negotiation may be the way to go about reaching distributed consensus range from aforementioned team robotics and unmanned vehicle applications [18, 24] to decentralized resource sharing in wireless communication networks [8] to enforcing consistency in distributed databases or data warehouses [3].

Consider an application such as a large ensemble of unmanned aerial vehicles (UAVs) deployed on a multi-task mission and spread across a sizable geographic area [5, 24]. Assume these UAVs are truly autonomous, and in particular not remotely controlled nor subject to centralized coordination. In such scenario, it is quite likely that different UAV agents will detect different tasks (for example, different regions or points of interest on which to perform surveillance), and hence strive to form different coalitions based on those tasks' estimated values and resource requirements [18]. Moreover, two different UAVs may perceive the same task differently – in terms of the expected utility from completing that task, estimated resource requirements for the task completion, etc. In such scenarios, even though all UAVs work on behalf of the same organization and are "team players" by design, they will still have different local preferences, and, consequently, may want to form different coalitions with their counterparts. The main lesson to be drawn from the general discussion, as well as the UAV example of a collaborative MAS where negotiation may be quite useful, is that sometimes *the differences in agents' perception and preferences*, and not necessarily their "selfishness", drive the need for negotiation. With that in mind, we turn to analyzing the MCDCF algorithm from a negotiation standpoint.

3.1 MCDCF from the Multi-Agent Negotiation Perspective

We outlined the generic problems of MAS coordination and cooperation, and then narrowed down our discussion to the more specific problem of distributed coalition formation. We then presented a distributed graph algorithm for multi-agent coalition formation, viewed as a coordination subroutine in a fully decentralized MAS setting. We subsequently motivated the usefulness and applicability of the negotiation approach to MAS distributed coalition formation – even when the agents aren't self-interested and can in general be expected to share their objectives. After discussing the problem of coalition formation from distributed graph algorithm design and MAS coordination viewpoints, we now discuss that algorithm from a negotiation perspective. We do so at two levels: (i) from the *rules of encounter*, that is, mechanism design perspective; and (ii) from the perspective of negotiation strategies for each individual agent [26]. We will discuss (i) in detail in the next subsection. To address (ii), we revisit how MCDCF works from a standpoint of a single agent, where, this time, the algorithm is viewed as a negotiation protocol.

We recall that multi-agent negotiation is in general comprised of three major components: the negotiation protocol, the object(s) of negotiation, and the agents' decision making models [6]. A negotiation protocol is a set of rules that govern the interaction among agents, and will be discussed in more detail shortly. Negotiation objects are a general concept capturing the range of issues over which agents need to reach agreement. In our context, there is only one such *object of negotiation*, namely, who will be forming a coalition with whom. Therefore, our setting as discussed in Section 2 is of the simplest, single object of negotiation nature, and this aspect of negotiation interaction need not be discussed further. Lastly, agents' decision making models have been defined as the decision-making apparatus that the agents employ in order to ensure acting in accordance with the negotiation protocol, so that they ensure satisfying their objectives (which, in our case, is to form coalitions). In general, (i) different agents may have different decision making models, and (ii) each agent's decision making model may depend on the protocol in place, the nature of object(s) of negotiation, and the range of operations that each agent is allowed to perform within the negotiation protocol [6]. In case of MCDCF, agents exchange coalition proposals until an agreement is reached. In particular, an agent engaging in a negotiation protocol based on MCDCF needs to be able to determine (a) whether it should stick to the current coalition proposal or change it to a different candidate coalition, (b) in case of the latter, which among possibly several available candidates for the new coalition proposal to adopt, and lastly (c) whether its coalition proposal has been accepted by the neighboring agents to whom that proposal was sent [22].

From a negotiation perspective, MCDCF is a distributed negotiation protocol that is carried out in several rounds. In each round, every agent, based on the history of the previous rounds and simple internal logic described in detail in [20–22] makes a coalition proposal to some nonempty subset of its neighbors. This coalition proposal may either be the same as what the agent proposed in the previous round, or it may be different; in case of the latter, the new coalition proposal is required to be *fresh*, i.e., to be a coalition that this agent has not proposed in any of the prior rounds. Whether an agent that has not joined a coalition yet ought to propose a new coalition or stick to its current choice is determined based on the internal logic that uses the *ChoiceFlag* and *NeighborFlag* information, as well as the proposals (and values of appropriate flags) received from the neighboring agents [19, 20]. We make the important observation that there is no explicit dependence on the messages (that is, coalition proposals) received from the neighbors during the previous rounds. That is, in each round, based on the proposals (and some additional information, captured by the appropriate flag values) received from the neighbors, an agent appropriately updates its internal state and, at the next round, updates the information about its neighbors. The decision flag is used to keep track, which among the neighbors have already joined a coalition in the prior rounds, and which are still engaging in negotiation. These properties ensure modest memory requirements of each agent: an agent only needs to keep a brief summary of the past rounds, not their detailed history.

In summary, each agent keeps negotiating from one round to the next, until eventually one of the following conditions is met:

– either a coalition proposal made to some of the agent's neighbors in a given round actually coincides with what each of those neighbors is proposing in that same round, in which case the agreement is detected and immediately established,

– or else, the agent has traversed its list of candidate coalitions and has arrived to the singleton coalition as the only still available option; in that case, the agent recognizes this "exit criterion", notifies the remaining neighbors (if any) of this situation, and forms a singleton coalition, whereupon it changes its *DecisionFlag* to 1 and exists the further negotiation.

In [21], we carefully prove that, under certain assumptions about sparseness of the underlying network topology, the amounts of communication, computation and storage per agent, per round of MCDCF are all reasonably modest. We have also established that the number of rounds is guaranteed to be finite regardless of the underlying topology, and polynomially bounded for sufficiently sparse network topologies.

3.2 The Mechanism Design View of MCDCF

We now turn attention to some desirable properties of negotiation protocols in general, and how well MCDCF viewed as a negotiation protocol for coalition formation measures up with respect to those properties. We first briefly review the basic concepts of the underlying theoretical framework for formulating desirable properties of negotiation protocols; that framework is provided by a sub-area of game theory, called mechanism design [6, 10, 25]. In the context of multi-agent systems, *mechanism design* addresses how to define the rules of multi-agent interaction in general, and of negotiation protocols, in particular [10]. Specifically, mechanism design for negotiation protocols defines the principles and policies so that, if a given negotiation protocol obeys them, and all the agents involved in negotiation "play by the rules" of the protocol, then certain properties can be guaranteed at the system level, regardless of the exact details of the negotiation, what kind of agreement (if any) is the negotiation going to result in a particular scenario, and the like. Following T. Sandholm in [25], we outline some of the highly desirable properties of a multi-agent negotiation protocol, and then discuss how well our MCDCF algorithm performs with respect to those properties.

- *Guaranteed success:* a negotiation algorithm holds this property if it ensures that, eventually (meaning, after finitely many negotiation rounds), an agreement is certain to be reached.
- *Pareto efficiency:* an outcome, O_p, of a negotiation protocol is *Pareto efficient* if there is no other outcome that would make one or more agents better off than in O_p, without making at least one other agent worse off.
- *Individual rationality:* a negotiation protocol is *individually rational* if following the protocol (as opposed to "cheating" or not engaging in it at all) is in the best interest of negotiation participants.

– *Stability:* a negotiation protocol is stable if all agents have an incentive to follow a particular strategy allowed by the protocol; that is, while there may be "legal" ways to deviate from a particular strategy, an agent that would choose such a way to deviate would, in general, be worse off than sticking to the stable strategy. In game theory, various notions of equilibria have been introduced to capture this notion of stability. The best known such notion is that of Nash equilibrium.
– *Simplicity:* a protocol is simple if each participant who is using the protocol can easily (i.e., computationally tractably) determine its stable or optimal strategy among the allowable strategies.
– *Distribution:* a protocol should be robust and fault-tolerant, esp. with respect to "single points of failure".

The *guaranteed success* property in our case amounts to guaranteed convergence after finitely many rounds. We have established the convergence properties of our MCDCF algorithm in [19, 20]. Hence, this property of MCDCF when viewed as a negotiation protocol holds. Similarly, since an agent cannot be worse off than remaining alone, engaging in coalition formation process results in at least as desirable an outcome as not engaging in coalition formation at all, and is therefore, in general, individually rational thing to do. (Discussion on how to properly evaluate, and possibly trade-off, the cost of engaging in MCDCF protocol in hope of joining some nontrivial coalition, versus saving oneself the trouble and the cost – and staying on one's own, is beyond our current scope; we do assume that the nature of tasks necessitates that agent join coalitions whenever possible.)

More interesting is *Pareto efficiency* of MCDCF. While we have not formally established Pareto optimality in our prior work, the simulations with the original, "baseline" version of MCDCF in practice always resulted in Pareto-optimal final coalition structures. However, the subsequent modifications and optimizations, especially those related to how each agent traverses a lattice of its candidate coalitions [22], while in general ensuring faster convergence once we scaled up our implementation from dozens to several hundreds of agents, also in certain cases may "hurt" the Pareto efficiency. Further modifying the algorithm so as to ensure Pareto efficiency, and then formally establishing that highly desirable property, are the subject of our ongoing work.

When it comes to *simplicity*, the MCDCF protocol is indeed fairly simple, and (unlike most negotiation protocols found in the literature) actually proven to be scalable to dozens or even hundreds of agents; this scalability holds under the single but critically important assumption of relative sparseness of the underlying graph, as discussed in earlier sections. We point out that this assumption indeed often holds in practice, and can be imposed by the system designer when it naturally does not hold, as we discuss in detail in [19–21]. Our protocol is very simple in terms of the communication language required for carrying out the negotiation process: the agents need to exchange tuples that encode the current coalition proposal and a few extra bits of auxiliary information (cf. the decision, choice and neighbor flags discussed earlier). For more details on the exact content of coalition proposal messages that agents exchange, see [21]. Lastly, as discussed earlier, MCDCF is a fully decentralized, *genuinely distributed* peer-to-peer algorithm; as such, when viewed as a negotiation protocol, it is robust to single points of failure and, in particular, satisfies the distribution property above.

4 Summary and Future Work

This paper has three main objectives: it (i) summarizes some aspects of our research on distributed coalition formation, (ii) motivates extending the negotiation paradigm from the traditional, game-theoretic *competitive* multi-agent domains to *collaborative*, distributed problem solving agents that are locally constrained and hence have different local preferences, and lastly (iii) discusses our coalition formation protocol and its properties from a negotiation standpoint. While the original motivation behind MCDCF was admittedly somewhat different, we have argued that our approach to coalition formation actually provides a simple and elegant negotiation protocol that has many desirable properties – and is demonstrably scalable to much larger collaborative MAS than what is commonly found in the existing game theoretic and negotiation-centric MAS literature. This scalability is to a considerable extent due to two particular properties of MCDCF: (1) it is a local algorithm, in that each agent only directly interacts with its immediate, one-hop neighbors; and (2) the messages exchanged during the course of the protocol execution are of a very simple nature.

Our plans for the future work include both some short-term, specific problems and a longer-term, broader research agenda. In the short term, we would like to determine whether our coalition formation protocol can be made Pareto efficient via relatively simple modifications to the coalition proposal exchange process. Insofar as the longer-term research agenda is concerned, we are interested in further expanding on various distributed and highly scalable negotiation protocols for coordination and cooperation in MAS, and investigating the potential benefits of *multi-tiered reinforcement learning* to more effective coordination and coalition formation.

Acknowledgments: This work was partially supported by NSF grant IIS 0914861.

References

1. S. Abdallah, V. Lesser. Organization-Based Cooperative Coalition Formation, in Proc. IEEE/WIC/ACM Int'l Conf. Intelligent Agent Technology, IAT'04 (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. J. Garcia-Garcia, C. Ordonez. Consistency-aware evaluation of OLAP queries in replicated data warehouses, in ACM DOLAP, pp. 73-80 (2009)
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, et al. An Actor-based Simulation for Studying UAV Coordination, in The 16th European Simulation Symposium (ESS'03), Delft, The Netherlands (2003)

6. N. Jennings, et al. Automated Negotiation: Prospects, Methods and Challenges, *in Group Decision & Negotiation*, vol. 10, pp. 199-215, Kluwer (2001)
7. 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)
8. L. Ma, X. Han, C.C. Shen. Dynamic open spectrum sharing MAC protocol for wireless ad hoc networks, in Proc. IEEE 1st Int'l Symp. on New Frontiers in Dynamic Spectrum Access Networks (DySPAN'05), pp. 203-213 (2005)
9. D. de Oliveira. Towards Joint Learning in Multiagent Systems Through Opportunistic Coordination, PhD Thesis, Univ. Fed. Do Rio Grande Do Sul, Brazil (2007)
10. S. Parsons, M. Wooldridge. Game Theory and Decision Theory in Multi-Agent Systems, in *Autonomous Agents and Multi-Agent Systems*, vol. 5, Kluwer (2002)
11. T.W. Sandholm, V.R. Lesser. Coalitions among computationally bounded agents, in Artificial Intelligence vol. 94, pp. 99-137 (1997)
12. 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 (1999)
13. O. Shehory, S. Kraus. Task allocation via coalition formation among autonomous agents, in Proceedings of IJCAI-95, Montreal, Canada, pp. 655-661 (1995)
14. O. Shehory, K. Sycara, S. Jha, Multi-agent coordination through coalition formation, in *Intelligent Agents IV: Agent Theories, Architectures & Languages*, LNAI vol. #1365, pp.153-164, Springer (1997)
15. O. Shehory, S. Kraus. Methods for task allocation via agent coalition formation, in *AI Journal*, vol. 101 (1998)
16. 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)
17. R. Sun. Meta-Learning Processes in Multi-Agent Systems, in *Intelligent agent technology: research & development*, N. Zhong, J. Liu (eds.), pp. 210-219, World Scientific, Hong Kong (2001)
18. P. Tosic, G. Agha. Understanding and modeling agent autonomy in dynamic multi-agent multi-task environments, in Proc. First European Workshop on Multi-Agent Systems (EUMAS'03) Oxford, England, UK (2003)
19. P. Tosic, G. Agha. Maximal Clique Based Distributed Group Formation Algorithm for Autonomous Agent Coalitions, in Proc. Workshop on Coalitions & Teams, within AAMAS'04, New York City, New York (2004)
20. P. Tosic, G. Agha. Maximal Clique Based Distributed Coalition Formation for Task Allocation in Large-Scale Multi-Agent Systems, Post-Proceedings of MMAS'04, LNAI vol. 3446, Springer (2005)
21. P. Tosic. Distributed Coalition Formation for Collaborative Multi-Agent Systems, MS thesis, Univ. of Illinois at Urbana-Champaign, Urbana, Illinois, USA (2006)
22. P. Tosic, N. Ginne. Some Optimizations in Maximal Clique based Distributed Coalition Formation for Collaborative Multi-Agent Systems, in Proc. IEEE Active Media Technology (AMT-2010), Toronto, Ontario, Canada (2010)
23. P. Tosic, N. Ginne. Challenges in Distributed Coalition Formation among Collaborative Multi-Agent Systems: An Experimental Case Study on Small-World Networks, in Proc. Int'l Conf. on Artif. Intelligence (ICAI'11), Las Vegas, Nevada, USA (2011)
24. S. Vig, J. A. Adams. Issues in multi-robot coalition formation, Proc. Multi-Robot Systems: From Swarms to Intelligent Automata, vol.3 (2005)
25. G. Weiss (ed.) *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, MIT Press (1999)
26. M. Wooldridge. *An Introduction to Multi-Agent Systems*, Wiley (2002)