Lecture 10

Outside of the traditional games

10.1 Games with a priori unions – a different interpretation of a coalition structure

So far, a coalition has represented a set of agents that worked on its own. In a CS, the different coalitions are intended to work independently of each other. We can also interpret a coalition to represent a group of agent that is more likely to work together within a larger group of agents (because of personal or political affinities). The members of a coalition do not mind working with other agents, but they want to be together and negotiate their payoff together, which may improve their bargaining power. This is the idea used in games with a priori unions. Formally, a game with a priori unions is similar to a game with CS: it consists of a triplet (N, v, S) when (N, v) is a TU game and S is a CS. However, we assume that the grand coalition forms. The problem is again to define a payoff distribution.

10.1.1. DEFINITION. [Game with a priori unions] A *game with a priori unions* is a triplet (N, v, S), where (N, v) is a TU game, and S is a particular CS. It is assumed that the grand coalition forms.

Owen [8] proposes a value that is based on the idea of the Shapley value. The agents forms the grand coalition by joining one by one. In the Shapley value, all possible joining orders are allowed. In the Owen value, an agent i may join only when the last agent that joined is a member of i's coalition or when the last agents $(j_1, \ldots, j_k)$ that joined before formed a coalition in S. This is formally captured using the notion of a consistency with a CS:

10.1.2. DEFINITION. [Consistency with a coalition structure] A permutation π is *consistent* with a CS S when, for all $(i, j) \in \mathcal{C}^2$, $\mathcal{C} \in S$ and $l \in N$, $\pi(i) < \pi(l) < \pi(j)$ implies that $l \in \mathcal{C}$.

We denote by $\Pi_S(N)$ the set of permutations of N that are consistent with the CS S. The number of such permutations is $m \prod_{\mathcal{C} \in S} |\mathcal{C}|!$ where m is the number of coalitions in S. The Owen value is then defined as follows:

10.1.3. DEFINITION. Owen value Given a game with a priori union (N, v, S), the Owen value $O_i(N, v, S)$ of agent i is given by

$$O_i(N, v, S) = \sum_{\pi \in \Pi_S(N)} \frac{mc(\pi)}{|\Pi_S(N)|}$$

In Table 10.1, we present the example used for the Shapley value and compute the Owen value. The members of the coalition of two agents improve their payoff by forming an union.

$$N = \{1, 2, 3\}$$
$$v(\{1\}) = 0 \quad v(\{2\}) = 0 \quad v(\{3\}) = 0$$
$$v(\{1,2\}) = 90 \quad v(\{1,3\}) = 80 \quad v(\{2,3\}) = 70$$
$$v(\{1,2,3\}) = 120$$

$S_2 = \{\{1,2\},\{3\}\}$

	1	2	3
$1 \leftarrow 2 \leftarrow 3$	0	90	30
$1 \leftarrow 3 \leftarrow 2$		✘	
$2 \leftarrow 1 \leftarrow 3$	90	0	30
$2 \leftarrow 3 \leftarrow 1$		✘	
$3 \leftarrow 1 \leftarrow 2$	80	40	0
$3 \leftarrow 2 \leftarrow 1$	50	70	0
total	220	200	60
Owen value $O_i(N, v, S_1)$	55	50	15

$S_2 = \{\{1,3\},\{2\}\}$

	1	2	3
$1 \leftarrow 2 \leftarrow 3$		✘	
$1 \leftarrow 3 \leftarrow 2$	0	40	80
$2 \leftarrow 1 \leftarrow 3$	90	0	30
$2 \leftarrow 3 \leftarrow 1$	50	0	70
$3 \leftarrow 1 \leftarrow 2$	80	40	0
$3 \leftarrow 2 \leftarrow 1$		✘	
total	220	80	180
Owen value $O_i(N, v, S_2)$	55	20	45

Table 10.1: Example of the computation of an Owen value

10.2 Games with externalities

A traditional assumption in the literature of coalition formation is that the value of a coalition depends solely on the members of that coalition. In particular, it is independent of on non-members' actions. In general, this may not be true: some externalities (positive or negative) can create a dependency between the value of a coalition and the actions of non-members. [10] attribute these externalities to the presence of shared resources (if a coalition uses some resource, they will not be available to other coalitions), or when there are conflicting goals: non-members can move the world farther from a coalition's goal state. [9] state that a "recipe for generating characteristic functions is a minimax argument": the value of a coalition $\mathcal{C}$ is the value $\mathcal{C}$ gets when the non-members respond optimally so as to minimise the payoff of $\mathcal{C}$. This formulation acknowledges that the presence of other coalitions in the population may affect the payoff of the coalition $\mathcal{C}$. As in [4, 9], we can study the interactions between different coalitions in the population: decisions about joining forces or splitting a coalition can depend on the way the competitors are organised. For example, when different companies are competing for the same market niche, a small company might survive against a competition of multiple similar individual small companies. However, if some of these small companies form a viable coalition, the competition significantly changes: the other small companies may now decide to form another coalition to be able to successfully compete against the existing coalition. Another such example is a bargaining situation where agents need to negotiate over the same issues: when agents form a coalition, they can have a better bargaining position, as they have more leverage, and because the other party needs to convince all the members of the coalition. If the other parties also form coalition, the bargaining power of the first coalition may decrease.

Two main types of games with externalities are described in the literature, both are represented by a pair (N, v), but the valuation function has a different signature.

Games in partition function form [11]: $v : 2^N \times \mathscr{S}_n \to \mathbb{R}$. This is an extension of the valuation function of a TU game by providing the value of a coalition given the current coalition structure (note that $v(\mathcal{C}, \mathcal{S})$ is meaningful when $\mathcal{C} \in \mathcal{S}$).

Games with valuations : $v : N \times \mathscr{S}_n \to \mathbb{R}$. In this type of games, the valuation function directly assigns a value to an agent given a coalition structure. One possible interpretation is that the problem of sharing the value of a coalition to the members has already been solved.

The definitions of superadditivity, subadditivity and monotonicity can be adapted to games in partition functions [3]. As an example, we provide the definition for superadditivity.

10.2.1. DEFINITION. [superadditive games in partition function] A partition function v is superadditive when, for any CS $\mathcal{S}$ and any coalitions $\mathcal{C}_1$ and $\mathcal{C}_2$ in $\mathcal{S}$, we have $v(\mathcal{C}_1 \cup \mathcal{C}_2, \mathcal{S} \setminus \{\mathcal{C}_1, C_2\} \cup \{\mathcal{C}_1 \cup \mathcal{C}_2\}) \geq v(\mathcal{C}1, \mathcal{S}) + v(B, \mathcal{S})$.

The partition function may also have some regularities when two coalition merge: either they always have a positive effect on the other coalition, or they always have a negative one. More precisely, a partition function exhibits *positive spillovers* when for any CS $\mathcal{S}$ and any coalitions $\mathcal{C}_1$ and $\mathcal{C}_2$ in $\mathcal{S}$, we have $v(\mathcal{C}, \mathcal{S} \setminus \{\mathcal{C}_1, C_2\} \cup \{\mathcal{C}_1 \cup \mathcal{C}_2\}) \geq v(\mathcal{C}, \mathcal{S})$ for all coalitions $\mathcal{C} \neq \mathcal{C}_1, \mathcal{C}_2$ in $\mathcal{S}$.

We now turn to considering solution concepts for such games. The issue of extending the Shapley value has a rich literature in game theory. We want the Shapley value to represent an average marginal contribution, but there is a debate over which set of coalition structures. Michalak et al. [5] provide references on different solutions and present three solutions in more details.

Airiau and Sen [2] considers the issue of the stability of the optimal CS and discusses a possible way to extend the kernel for partition function games. In [1], they consider coalition formation in the context of games with valuations and propose a solution for myopic agents (an agent will join a coalition only when it is beneficial, without considering long-terms effect).

Michalak et al. [7] tackle the problem of representing such games and propose three different representations that depends on the interpretation of the externalities. The first representation considers the value of a coalition in a CS: the value of a coalition can be decomposed into on term that is free of externality and another term that models the sum of the uncertainty due to the formation of the other coalitions. The two other representations consider that the contribution of a coalition in a CS: either by providing the mutual influence of any two coalitions in a CS (outward operational externalities) or by providing the influence of all the other coalitions on a given coalition (inward operational externalities). Michalak et al. (in [5] and [6]) extend the concept of MC-nets to games with partition function.

Bibliography

[1] Stéphane Airiau and Sandip Sen. A fair payoff distribution for myopic rational agents. In *Proceedings of the Eighth International Conference on Autonomous Agents and Multiagent Systems (AAMAS-09)*, May 2009.

[2] Stéphane Airiau and Sandip Sen. On the stability of an optimal coalition structure. In *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI-2010)*, pages 203–208, August 2010.

[3] Francis Bloch. Non-cooperative models of coalition formation in games with spillover. In Carlo Carraro, editor, *The endogenous formation of economic coalitions*, chapter 2, pages 35–79. Edward Elgar, 2003.

[4] Sergiu Hart and Mordecai Kurz. Endogenous formation of coalitions. *Econometrica*, 51(4), July 1983.

[5] Tomasz Michalak, Talal Rahwan, Dorota Marciniak, Marcin Szamotulski, and Nicholas R. Jennings. Computational aspects of extending the shapley value to coalitional games with externalities. In *Proceeding of the 2010 conference on ECAI 2010: 19th European Conference on Artificial Intelligence*, pages 197–202, Amsterdam, The Netherlands, The Netherlands, 2010. IOS Press.

[6] Tomasz Michalak, Talal Rahwan, Dorota Marciniak, Marcin Szamotulski, Peter McBurney, and Nicholas R. Jennings. A logic-based representation for coalitional games with externalities. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1 - Volume 1*, AAMAS '10, pages 125–132, Richland, SC, 2010. International Foundation for Autonomous Agents and Multiagent Systems.

[7] Tomasz Michalak, Talal Rahwan, Jacek Sroka, Andrew Dowell, Michael Wooldridge, Peter McBurney, and Nicholas R. Jennings. On representing coalitional games with externalities. In *Proceedings of the 10th ACM conference on Electronic Commerce 09 (EC'09)*, 2009.

[8] Guilliermo Owen. Values of games with a priori unions. In O. Moeschlin R. Hein, editor, *Mathematical Economics and Game Theory: Essays in Honor of Oskar Morgenstern*. Springer, New York, 1977.

[9] Debraj Ray and Rajiv Vohra. A theory of endogenous coalition structures. *Games and Economic Behavior*, 26:286–336, 1999.

[10] Tuomas Sandholm and Victor R. Lesser. Coalitions among computationally bounded agents. *AI Journal*, 94(1–2):99–137, 1997.

[11] R. M. Thrall and W. F. Lucas. N-person games in partition function form. *Naval Research Logistics Quarterly*, 10(1):281–298, 1963.

Lecture 11
Coalition Structure Generation problem and related issues

In the previous sections, the focus was on individual agents that are concerned with their individual payoff. In this section, we consider TU games (N, v) in which agents are concerned only about the society's payoff: the agents' goal is to maximise utilitarian social welfare. The actual payoff of the agent or the value of her coalition is not of importance in this setting, only the total value generated by the population matters. This is particularly interesting for multiagent systems designed to maximise some objective functions. In the following, an optimal CS denotes a CS with maximum social welfare. This may model multiagent systems that are designed to optimise an objective function.

More formally, we consider a TU game (N, v), and we recall that a coalition structure (CS) $s = \{\mathcal{S}_1, \cdots, \mathcal{S}_m\}$ is a partition of N, where $\mathcal{S}_i$ is the i^{th} coalition of agents, and $i \neq j \Rightarrow \mathcal{S}_i \cap \mathcal{S}_j = \emptyset$ and $\cup_{i\in[1..m]}\mathcal{S}_i = N$. $\mathscr{S}$ denotes the set of all CSs. The goal of the multiagent system is to locate a CS that maximises utilitarian social welfare, in other words the problem is to find an element of $\text{argmax}_{s\in\mathscr{S}} \sum_{\mathcal{S}\in s} v(\mathcal{S})$.

The space $\mathscr{S}$ of all CSs can be represented by a lattice, and an example for a population of four agents is provided in Figure 11.1. The first level of the lattice consists only of the CS corresponding to the grand coalition $N = \{1, 2, 3, 4\}$, the last level of the lattice contains CS containing singletons only, i.e., coalitions containing a single member. Level i contains all the CSs with exactly i coalitions. The number of CSs at level i is $\mathcal{S}(|N|, i)$, where $\mathcal{S}$ is the Stirling Number of the Second Kind[1]. The Bell number, $\mathscr{B}(n)$, represents the total number of CSs with n agents, $\mathscr{B}(n) = \sum_{i=0}^{n} \mathcal{S}(n, k)$. This number grows exponentially, as shown in Figure 11.2, and is $O(n^n)$ and $\omega(n^{\frac{n}{2}})$ [15]. When the number of agents is relatively large, e.g., $n \geq 20$, exhaustive enumeration may not be feasible.

The actual issue is the search of the optimal CS. Sandholm et al. [15] show that given a TU game (N, v), the finding the optimal CS is an $\mathcal{NP}$-complete problem. In the following, we will consider centralised search where a single agent is performing

[1] $\mathcal{S}(n, m)$ is the number of ways of partitioning a set of n elements into m non-empty sets.

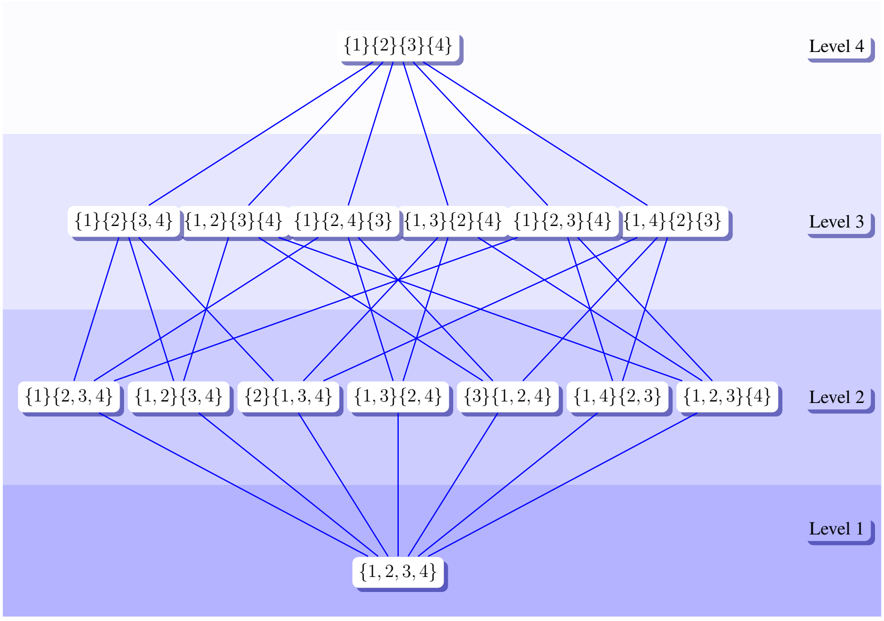

Figure 11.1: Set of CSs for 4 agents.

the search as well as the more interesting case of decentralised search where all agents make the search at the same time on different parts of the search space. Before doing so, we review some work where the valuation function v is not known in advance. In a real application, these values need to be computed; and this may be an issue on its own if the computations are hard, as illustrated by an example in [14] where the computation of a value requires to solve a traveling salesman problem.

11.1 Sharing the computation of the coalition values

Thus far, when we used a TU game, the valuation function was common knowledge. For a practical problem though, one needs to compute these values. We said that the value of a coalition was the worth that could be achieved through cooperation of the coalition's members. In many cases, computing the value of a coalition will be an optimization problem: find the optimal way to cooperate to produce the best possible worth. In some cases, such a problem may be computationaly hard. The following example is given by Sandholm and Lesser [14]: we are in a logistics application and the computing the value of a coalition requires to solve a travelling salesman problem, a problem known to be $\mathcal{NP}$-complete. Before being able to compute an optimal CS, one needs to compute the value of all coalitions. Since agents are cooperative (i.e. they want to work together to ensure the best outcome for the society), we are interested in a decentralised algorithm that computes all the coalition values in a minimal amount of time, and that requires minimum communication between the agents.

Shehory and Kraus were the first to propose an algorithm to share the computation

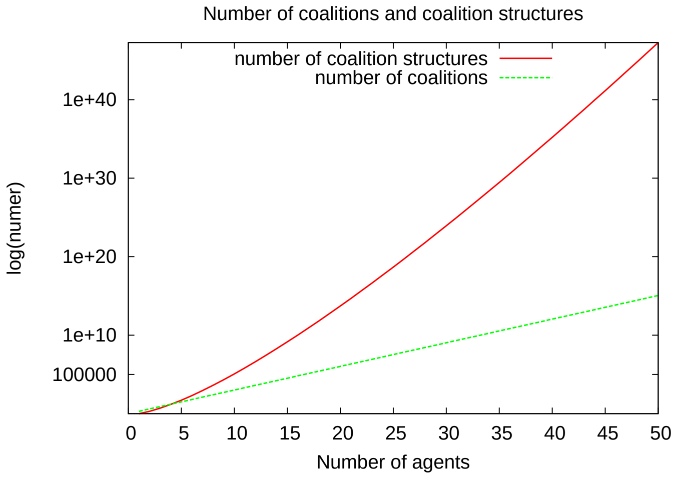

Figure 11.2: Number of CSs in a population of n agents.

of the coalition values [19]. In their algorithm, the agents negotiate which computation is performed by which agent, which is quite demanding. Rahwan and Jennings proposed an algorithm where agents first agree on an identification for each agent participating in the computation (an index between 1 and n the number of agents). Then, each agent use the same algorithm that determines which coalition values they need to compute, removing the need of any further communication, except announcing the result of the computation. The index is used to compute a set of coalitions and ensures that the values of all the coalitions are computed exactly once. This algorithm, called DCVC [7] outperforms the one by Shehory and Kraus. To minimize the overall time of computation, it is best to balance the work of all the agents. The key observation is that in general, it should take longer to compute the value of a large coalition compared to a small coalition (i.e., the computational complexity is likely to increase with the size of the coalition since more agents have to coordinate their activities). Their method improves the balance of the loads by distributing coalitions of the same size to all agents. By knowing the number of agents n participating in the computation an index number (i.e., an integer in the range $\{0..n\}$), the agents determine for each coalition size which coalition values to compute. The algorithm can also be adapted when the agents have different known computational speed so as to complete the computation in a minimum amount of time.

11.2 Searching for the optimal coalition structure

Once the value of each coalition is known, the agents needs to search for an optimal CS. The difficulty of this search lies in the large search space, as recognised by existing algorithms, and this is even more true in the case where there exists externalities (i.e., when the valuation of a coalition depends on the CS). For TU games with no externalities, some algorithms guarantee finding CSs within a bound from the optimum when an incomplete search is performed. Unfortunately, such guarantees are not possible for games with externalities. We shortly discuss these two cases in the following.

11.2.1 Games with no externalities

Anytime algorithms

Sandholm et al. [15] proposed a first algorithm that searches through a lattice as presented in Figure 11.1. Their algorithm guarantees that the CS found, s, is within a bound from the optimal $s^\star$ when a sufficient portion of the lattice has been visited. To ensure any bound, it is necessary to visit a least 2^{n-1} CSs (Theorems 1 and 3 in [15]) which corresponds to the first two levels of the lattice, i.e., the algorithm needs to visit the grand coalition and all the CSs composed of 2 coalitions. Let $\mathcal{S}'$ be the best CS found in the first two levels, then we have $v(s^\star) \leq n \cdot v(\mathcal{S}')$. To see this, let $\mathcal{C}_{max}$ a coalition with the highest value (i.e. $\mathcal{C}_{max} \in \text{argmax}_{\{\mathcal{C}\subseteq N\}} v(\mathcal{C})$. It is clear that $v(s^\star) \leq n \times v(\mathcal{C}_{max})$ as each coalition forming the CS $s^\star$ has a most the value of $v(\mathcal{C}_{max})$ and there are at most n coalitions in $s^\star$. Since all coalitions are part of these levels, it is clear that we have $v(\mathcal{C}_{max}) \leq v(\mathcal{S}')$. Finally, we have $v(s^\star) \leq n \times v(\mathcal{S}')$, which was what we wanted.

The bound improves each time a new level is visited. An empirical study of different strategies for visiting the other levels is presented in [4]. Three different algorithms are empirically tested over characteristic functions with different properties: 1) subadditive, 2) superadditive, 3) picked from a uniform distribution in $[0, 1]$ or in $[0, |\mathcal{S}|]$ (where $|\mathcal{S}|$ is the size of the coalition). The performance of the heuristics differs over the different type of valuation functions, demonstrating the importance of the properties of the characteristic function in the performance of the search algorithm.

The algorithm by Dang and Jennings [3] improves the one of [15] for low bounds from the optimal. For large bounds, both algorithms visit the first two levels of the lattice. Then, when the algorithm by Sandholm et al. continues by searching each level of the lattice, the algorithm of Dang and Jennings only searches specific subset of each level to decrease the bound faster. This algorithm is anytime, but its complexity is not polynomial.

These algorithms were based on a lattice as the one presented in Figure 11.1 where a CS in level i contains exactly i coalitions. The best algorithm to date has been developed by Rahwan et al. and uses a different representation called integer-partition (IP) of the search space. It is an anytime algorithm that has been improved over a series of paper: [11, 12, 8, 9, 13]. In this representation the CSs are grouped according to the sizes of the coalitions they contain, which is called a configuration. For example, for a population of four agents, the configuration $\{1, 3\}$ represents CSs that contain a coalition with a singleton and a coalition with three agents. A smart scan of the input allows to search the CSs with two coalitions the grand coalition and the CS containing singletons only. In addition, during the scan, the algorithm computes the average and maximum value for each coalition size. The maximum values can be used to prune the search space. When constructing a configuration, the use of the maximum values of a coalition for each size permits the computation of an upper bound of the value of a CS that follows that configuration, and if the value is not greater than the current best CS, it is not necessary to search through the CSs with that configuration, which prunes the search tree. Then, the algorithm searches the remaining configurations, starting with the most promising ones. During the search of a configuration, a branch and bound technique is used. In addition, during the search, the algorithm is designed so that no CS is evaluated twice. Empirical evaluation shows that the algorithm outperforms any other current approach over different distributions used to generate the values of the coalitions.

dynamic programming

Another approach is to use dynamic programming technique. The key idea is provided in the following lemma: in order to compute the optimal value of a CS, it suffices to consider partitions of N into two disjoints coalitions and apply the argument recursively. To help us, let us recall the definition of the supeadditive cover $(N, \hat{v})$ of a TU game (N, v). The valuation function $\hat{v}$ is $\hat{v}(\mathcal{C}) = \max_{\mathcal{P} \in \mathscr{S}_{\mathcal{C}}} \left\{ \sum_{T \in \mathcal{P}} v(T) \right\}$ for all $\mathcal{C} \subseteq N \setminus \emptyset$ and $\hat{v}(\emptyset) = 0$. The set of optimal CSs can now be noted argmax $\hat{v}(N)$. Let us now state the key lemma:

11.2.1. LEMMA. *For any $\mathcal{C} \subseteq N$, we have*

$$\hat{v}(\mathcal{C}) = \max \left\{ \max \left\{ \hat{v}(\mathcal{C}') + \hat{v}(\mathcal{C}'') \mid \mathcal{C}' \cup \mathcal{C}'' = \mathcal{C} \wedge \mathcal{C}' \cap \mathcal{C}'' = \emptyset \wedge \mathcal{C}', \mathcal{C}'' \neq \emptyset \right\}, v(\mathcal{C}) \right\}.$$

Proof. Clearly, $\hat{v}(\mathcal{C}) \geq v(C)$. Take two disjoint non-empty coalitions $\mathcal{C}'$ and $\mathcal{C}''$ such that $\mathcal{C}' \cup \mathcal{C}'' = \mathcal{C}$. Let $\mathcal{S}'$ and $\mathcal{S}''$ be two partitions of $\mathcal{C}'$ and $\mathcal{C}''$ such that $\hat{v}(\mathcal{C}') = v(\mathcal{S}')$ and $\hat{v}(\mathcal{C}'') = v(\mathcal{S}'')$. Then $\mathcal{S}' \cup \mathcal{S}''$ is a CS over $\mathcal{C}$ with $v(\mathcal{S}' \cup \mathcal{S}'') = v(\mathcal{S}') + v(\mathcal{S}'')$, so we must have $\hat{v}(\mathcal{C}) \geq \hat{v}(\mathcal{C}') + v(\mathcal{C}'')$.

Now, let $\mathcal{S}$ be a partition of $\mathcal{C}$ such that $\hat{v}(\mathcal{C}) = v(\mathcal{S})$. If $\mathcal{S} = \{\mathcal{C}\}$, then we are done. Otherwise, let $\mathcal{C}'$ be a coalition in $\mathcal{S}$, $\mathcal{C}'' = C \setminus \mathcal{C}'$ and $\mathcal{S}'$ be $\mathcal{S} \setminus \{\mathcal{C}'\}$. Since $\mathcal{S}'$ is a CS over $\mathcal{C}''$, we have $\hat{v}(\mathcal{C}'') \geq v(\mathcal{S}') = v(\mathcal{S}) - v(\mathcal{C}')$. On the other hand, we have $\hat{v}(\mathcal{C}') \geq v(\mathcal{C}')$. Hence $\hat{v}(\mathcal{C}') + \hat{v}(\mathcal{C}'') \geq v(\mathcal{S}) = \hat{v}(\mathcal{C})$. □

More recently, [17, 18] designed an algorithm that uses dynamic programming and that guarantees a constant factor approximation ratio r in a given time. In particular, the latest algorithm [17] guarantees a factor of $\frac{1}{8}$ in $O(2^n)$.

Other approaches

Some algorithms are now trying to combine an anytime approach and an dynamics programming. Other researchers try to use different techniques. For example, Silaghi et al [20] propose to use a different representation, assuming that the value of a coalition is the optimal solution of a distributed constraint optimization problem (DCOP). The algorithm uses a DCOP solver and guarantees a bound from the optimum.

The algorithms above assume that the TU game is represented in a naive way. There exists some algorithms that take advantage of compact representation. For example, [6] proposes algorithms in the case where the game is represented using an MC-nets and in the case where the synergy coalition group is used. Another example is [1] for skill games.

11.2.2 Games with externalities

The previous algorithm explicitly uses the fact that the valuation function only depends on the members of the coalition, i.e., has no externalities. When this is not the case, i.e., when the valuation function depends on the CS, it is still possible to use some algorithms, e.g., the one proposed in [4], but the guarantee of being within a bound from the optimal is no longer valid. Sen and Dutta use genetic algorithms techniques [16] to perform the search. The use of such technique only assumes that there exists some underlying patterns in the characteristic function. When such patterns exist, the genetic search makes a much faster improvement in locating higher valued CS compared to the level-by-level search approach. One downside of the genetic algorithm approach is that there is no optimality guarantee. Empirical evaluation, however, shows that the genetic algorithm does not take much longer to find a solution when the value of a coalition does depend on other coalitions.

More recently, Rahwan et al. and Michalak et al. consider the problem for some class of externalities and modify the IP algorithm for the games with externalities [5, 10], however, they assume games with negative or positive spillovers. [2] introduce a representation to represent games in partition function games using types: each agent has a single type. They make two assumptions on the nature of the externalities (based on the notions of competition and complementation) and they show that games with negative or positive spillovers are special cases. They provide a branch and bound algorithm for the general setting. They also provide a worst-case initial bound.

Bibliography

[1] Yoram Bachrach, Reshef Meir, Kyomin Jung, and Pushmeet Kohli. Coalitional structure generation in skill games. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10)*, pages 703–708, July 2010.

[2] Bikramjit Banerjee and Landon Kraemer. Coalition structure generation in multi-agent systems with mixed externalities. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1 - Volume 1*, AAMAS '10, pages 175–182, Richland, SC, 2010. International Foundation for Autonomous Agents and Multiagent Systems.

[3] Viet Dung Dang and Nicholas R. Jennings. Generating coalition structures with finite bound from the optimal guarantees. In *Proceedings of the third International Joint Conference on Autonomous Agents and Multiagent Systems(AAMAS'04)*, 2004.

[4] Kate S. Larson and Tuomas W. Sandholm. Anytime coalition structure generation: an average case study. *Journal of Experimental & Theoretical Artificial Intelligence*, 12(1):23–42, 2000.

[5] Tomasz Michalak, Andrew Dowell, Peter McBurney, and Michael Wooldridge. Optimal coalition structure generation in partition function games. In *Proceeding of the 2008 conference on ECAI 2008*, pages 388–392, Amsterdam, The Netherlands, The Netherlands, 2008. IOS Press.

[6] Naoki Ohta, Vincent Conitzer, R. Ichimura, Y. Sakurai, and Makoto Yokoo. Coalition structure generation utilizing compact characteristic function representations. In *Proocedings of the 15th International Conference on Principles and-Practice of Constraint Programming (CP'09)*, pages 623–638, 2009.

[7] Talal Rahwan and Nicholas R. Jennings. An algorithm for distributing coalitional value calculations among cooperating agents. *Artificial Intelligence*, 171(8-9):535–567, 2007.

[8] Talal Rahwan and Nicholas R. Jennings. Coalition structure generation: Dynamic programming meets anytime optimization. In *Proceedings of the 23rd conference on artificial intelligence (AAAI-08)*, pages 156–161, 2008.

[9] Talal Rahwan and Nicholas R. Jennings. An improved dynamic programming algorithm for coalition structure generation. In *Proceedings of the 7th international conference on Autonomous Agents and Multi-Agent Systems (AAMAS-08)*, 2008.

[10] Talal Rahwan, Tomasz Michalak, Nicholas R. Jennings, Michael Wooldridge, and Peter McBurney. Coalition structure generation in multi-agent systems with positive and negative externalities. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI-09)*, 2009.

[11] Talal Rahwan, Sarvapali D. Ramchurn, Viet Dung Dang, Andrea Giovannucci, and Nicholas R. Jennings. Anytime optimal coalition structure generation. In *Proceedings of the Twenty-Second Conference on Artificial Intelligence (AAAI-07)*, pages 1184–1190, 2007.

[12] Talal Rahwan, Sarvapali D. Ramchurn, Viet Dung Dang, and Nicholas R. Jennings. Near-optimal anytime coalition structure generation. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI'07)*, pages 2365–2371, January 2007.

[13] Talal Rahwan, Sarvapali D. Ramchurn, Nicholas R. Jennings, and Andrea Giovannucci. An anytime algorithm for optimal coalition structure generation. *Journal of Artificial Intelligence Research*, 34:521–567, 2009.

[14] Tuomas Sandholm and Victor R. Lesser. Coalitions among computationally bounded agents. *AI Journal*, 94(1–2):99–137, 1997.

[15] Tuomas W. Sandholm, Kate S. Larson, Martin Andersson, Onn Shehory, and Fernando Tohmé. Coalition structure generation with worst case guarantees. *Artificial Intelligence*, 111(1–2):209–238, 1999.

[16] Sandip Sen and Partha Sarathi Dutta. Searching for optimal coalition structures. In *ICMAS '00: Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS-2000)*, page 287, Washington, DC, USA, 2000. IEEE Computer Society.

[17] Travis Service and Julie Adams. Approximate coalition structure generation. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10)*, pages 854–859, July 2010.

[18] Travis Service and Julie Adams. Constant factor approximation algorithms for coalition structure generation. *Autonomous Agents and Multi-Agent Systems*, pages 1–17, 2010. Published online February 2010.

[19] Onn Shehory and Sarit Kraus. Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101(1-2):165–200, May 1998.

[20] Suguru Ueda, Atsushi Iwasaki, Makoto Yokoo, Marius Calin Silaghi, Katsutoshi Hirayama, and Toshihiro Matsui. Coalition structure generation based on distributed constraint optimization. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10)*, pages 197–203, July 2010.

Lecture 12
Issues for applying cooperative games

We now highlight issues that have emerged from the different types of applications (e.g. resource or task allocation problem or forming a buying group). Some of the issues have solutions while others remain unsolved, for example, dealing with agents that can enter and leave the environment at any time in an open, dynamic environment. None of the current protocols can handle these issues without re-starting computation, and only few approaches consider how to re-use the already computed solution [6, 13].

12.1 Stability and Dynamic Environments

Real-world scenarios often present dynamic environments. Agents can enter and leave the environment at any time, the characteristics of the agents may change with time, the knowledge of the agents about the other agents may change, etc.

The game-theoretic stability criteria are defined for a fixed population of agents and the introduction of a new agent in the environment requires significant computation to update a stable payoff distribution. For example, for the kernel, all the agents need to check whether any coalition that includes the new agent changes the value of the maximum surplus, which requires re-evaluating $O(2^n)$ coalitions. Given the complexity of the stability concept, one challenge that is faced by the multiagent community is to develop stability concepts that can be easily updated when an agent enters or leaves the environment.

In addition, if an agent drops during the negotiation, this may cause problems for the remaining agents. For example, a protocol that guarantees a kernel stable payoff distribution is shown not to be 'safe' when the population of agents is changing: if an agent i leaves the formation process without notifying other agents, the other agents may complete the protocol and find a solution to a situation that does not match the reality. Each time a new agent enters or leaves the population, a new process needs to be restarted [9].

In an open environments, manipulations will be impossible to detect: agents may use multiple identifiers (or false names) to pretend to be multiple agents, or the other way around, multiple agents may collude and pretend to be a single agents, or agents can hide some of their skills. Hence, it is important to propose solution concepts that are robust against such manipulations. We will come back later to some of the solution that have been proposed: the anonymity-proof core [44] and anonymity-proof Shapley value [35].

12.2 Uncertainty about Knowledge and Task

In real-world scenario, agents will be required to handle some uncertainty. Different sources of uncertainty have been considered in the literature:

- the valuation function is an approximation [38] and agents may not use the same algorithm. Hence, the agents may not know what is the true value.
- agents may not know some tasks [9] or the value of some coalitions. In such cases, the agents play a different coalitional game that may reduce the payoff of some agents compared to the solution of the true game.
- some information is private, i.e., an agent knows some property about itself, but does not know it for other agents. In [28], it is the cost incurred by other agents to perform a task that is private. In [16, 17], agents have a private type, and the valuation function depends on the types of the coalition's members.
- uncertainty about the outcome of an action [16]: when a coalition makes an action, some external factors may influence the outcome of the actions. This can be captured by a probability of an outcome given the action taken and the type of the members of the coalition.
- there are multiple possible worlds [24], which models the different possible outcomes of the formation of a coalition. Agents know a probability distribution over the different worlds. In addition, an agent may not able to distinguish some worlds as it lacks information and they know a partition of the worlds (called information sets), each set of the partition represent worlds that appears as indistinguishable.

Some authors also consider that there is uncertainty in the valuation function without modeling a particular source, for example in [25], each agent has an expectation of the valuation function. In [10, 11] fuzzy sets are used to represent the valuation function. In the first paper, the agents enters bilateral negotiations to negotiate Shapley value, in the second paper, they define a fuzzy version of the kernel.

In the uncertainty model of [24], the definition of the core depends on the time one reasons about it. They proposed three different definitions of the core that depend on the timing of the evaluation: before the world is drawn or *ex-ante*, not much information can be used; after the world is drawn but before it is known, also called *ex-interim*, an agent knows to which set of its information set the real world belongs, but does not know which one; finally when the world is announced to the agent or *ex-post*, everything is known.

The model of [16] combines uncertainty about the agent types and uncertainty about the outcome of the action taken by the coalition. Each agent has a probabilistic belief about the types of the other agents in the population. Chalkiadakis and Boutilier propose a definition of the core, the Bayesian core (introduced in [14]) in which no agent has the belief that there exists a better coalition to form. As it may be difficult to obtain all the probabilities and reason about them, [17] propose to use a "point" belief: an agent guesses the type of the other agents and reason with these guesses. The paper analyses the core, simple games (proving that the core of a simple game is non-empty iff the game has a veto player) and some complexity result in this games with belief.

12.3 Safety and Robustness

It is also important that the coalition formation process is robust. For instance, communication links may fail during the negotiation phase. Hence, some agents may miss some components of the negotiation stages. This possibility is studied in [9] for the KCA protocol [27]: coalition negotiations are not safe when some agents become unavailable (intentionally or otherwise). In particular, the payoff distribution is not guaranteed to be kernel-stable. [6] empirically studies the robustness of the use of a central algorithm introduced in [5]: the cost to compute a task allocation and payoff distribution in the core is polynomial, but it can still be expensive. In the case of agent failure, the computation needs to be repeated. Belmonte et al. propose an alternative payoff division model that avoids such a re-computation, but the solution is no longer guaranteed to be in the core, it is only close to the core. There is a trade-off between computational efficiency and the utility obtained by the agent. They conclude that when the number of agents is small, the loss of utility compared to the optimal is small; hence, the improvement of the computational efficiency can be justified. For a larger number of agents, however, the loss of utility cannot not justify the improvement in computational cost.

12.3.1 Protocol Manipulation

When agents send requests to search for members of a coalition or when they accept to form a coalition, the protocol may require disclosure of some private information [36]. When the agents reveal some of their information, the mechanism must ensure that there is no information asymmetry that can be exploited by some agents [7]. To protect a private value, some protocol [9] may allow the addition of a constant offset to the private value, as long as this addition does not impact the outcome of the negotiation.

Belmonte et al. study the effect of deception and manipulation of their model in [6]. They show that some agents can benefit from falsely reporting their cost. In some other approaches [9, 20], even if it is theoretically possible to manipulate the protocol, it is not possible in practice as the computational complexity required to ensure higher outcome to the malevolent agent is too high. For example, [20] show that manipulating marginal-contribution based value division scheme is $\mathcal{NP}$-hard (except when the valuation function has other properties, such as being convex).

Other possible protocol manipulations include hiding skills, using false names, colluding, etc. The traditional solution concepts can be vulnerable to false names and to collusion [44]. To address these problems, it is beneficial to define the valuation function in terms of the required skills instead of defining it over the agents: only skills, not agents, should be rewarded by the characteristic function. In that case, the solution concept is robust to false names, collusion, and their combination. But the agents can have incentive to hide skills. A straight, naive decomposition of the skills will increase the size of the characteristic function, and [45] propose a compact representation in this case.

12.4 Communication

While one purpose of better negotiation techniques may be to improve the quality of the outcome for the agents, other goals may include decreasing the time and the number of messages required to reach an agreement. For example, learning is used to decrease negotiation time in [41]. The motivation Lerman's work in [30] is to develop a coalition formation mechanism that has low communication and computation cost. In another work, the communication costs are included in the characteristic function [42].

The communication complexity of some protocols has been derived. For instance, the exponential protocol in [40] and the coalition algorithm for forming Bilateral Shapley Value Stable coalition in [26] have communication complexity of $O(n^2)$, the negotiation based protocol in [40] is $O(n^2 2^n)$, and it is $O(n^k)$ for the protocol in [39] (where k is the maximum size of a coalition). The goal of [37] is to analyse the communication complexity of computing the payoff of a player with different stability concepts: they find that it is $\Theta(n)$ when either the Shapley value, the nucleolus, or the core are used.

12.5 Scalability

When the population of heterogeneous agents is large, discovering the relevant agents to perform a task may be difficult. In addition, if all agents are involved in the coalition formation process, the cost in time and computation will be large. To alleviate this scalability issue, a hierarchy of agents can be used [1]. When an agent discovers a task that can be addressed by agents below this agent in the hierarchy, the agent picks the best of them to perform the task. If the agents below cannot perform the task, the agent passes the task to the agent above it in the hierarchy and the process repeats. The notion of clans [22] and congregations [12], where agents gather together for a long period have been proposed to restrict the search space by considering only a subset of the agents (see Section 12.6).

12.6 Long Term vs. Short Term

In general, a coalition is a short-lived entity that is "formed with a purpose in mind and dissolve when that need no longer exists, the coalition ceases to suit its designed purpose, or critical mass is lost as agents depart" [23]. It can be beneficial to consider the formation of long term coalitions, or the process of repeated coalition formation involving the same agents. [43] explicitly study long term coalitions, and in particular the importance of trust in this content. [12] refer to a long term coalition as a congregation. The purpose of a congregation is to reduce the number of candidates for a successful interaction: instead of searching the entire population, agents will only search in the congregation. The goal of a congregation is to gather agents, with similar or complementary expertise to perform well in an environment in the long run, which is not very different from a coalition. The only difference is that group rationality is not expected in a congregation. The notion of congregation is similar to the notion of clans [22]: agents gather not for a specific purpose, but for a long-term commitment. The notion of trust is paramount in the clans, and sharing information is seen as another way to improve performance.

12.7 Fairness

Stability does not necessarily imply fairness. For example, let us consider two CSs $\mathcal{S}$ and $\mathcal{T}$ with associated kernel-stable payoff distribution $x_{\mathcal{S}}$ and $x_{\mathcal{T}}$. Agents may have different preferences between the CSs. It may even be the case that there is no CS that is preferred by all agents. If the optimal CS is formed, some agents, especially if they are in a singleton coalition, may suffer from the choice of this CS. [3] propose a modification of the kernel to allow side-payment between coalitions to compensate such agents.

[2] consider partition function games with externalities. They consider a process where, in turns, agents change coalition to improve their immediate payoff. They propose that the agents share the maximal social welfare, and the size of the share is proportional to the expected utility of the process. The payoff obtained is guaranteed to be at least as high as the expected utility. They claim that using the expected utility as a base of the payoff distribution provides some fairness as the expected utility can be seen as a global metric of an agent performance over the entire set of possible CSs.

12.8 Overlapping Coalitions

It is typically assumed that an agent belongs to a single coalition; however, there are some applications where agents can be members of multiple coalitions. For instance, the expertise of an agent may be required by different coalitions at the same time, and the agent can have enough resources to be part of two or more coalitions. In a traditional setting, the use of the same agent i by two coalitions $\mathcal{C}_1$ and $\mathcal{C}_2$ would require a merge of the two coalitions. This larger coalition U is potentially harder to manage, and a priori, there would not be much interaction between the agents in $\mathcal{C}_1$ and $\mathcal{C}_2$, except for agent i. Another application that requires the use of overlapping coalition is tracking targets using a sensor networks [21]. In this work, a coalition is defined for a target, and as agents can track multiple targets at the same time, they can be members of different coalitions.

The traditional stability concepts do not consider this issue. One possibility is for the agent to be considered as two different agents, but this representation is not satisfactory as it does not capture the real power of this agent. Shehory and Kraus propose a setting with overlapping coalition [39]: Each agent has a capacity, and performing a task may use only a fraction of the agent's capacity. Each time an agent commits to a task, the possible coalitions that can perform a given task can change. A mapping to a set covering problem allows to find the coalition. However, the study of the stability is not considered. Another approach is the use of fuzzy coalition [8]: agents can be members of a coalition with a certain degree that represents the risk associated with being in that coalition. Other work considers that the agents have different degree of membership, and their payoff depends on this degree [4, 31, 34]. The protocols in [29] also allow overlapping coalitions.

More recently, [19][1] have studied the notion of the core in overlapping coalition formation. In their model, each agent has one resource and the agent contributes a fraction of that resource to each coalition it participates in. The valuation function v is then $[0,1]^n \rightarrow \mathbb{R}$. A CS is no longer a partition of the agents: a CS S is a finite list of vectors, one for each 'partial' coalition, i.e., $S = (r^1, \ldots, r^k)$. The size of S is the number of coalitions, i.e., k. The support of $r^{\mathcal{C}} \in S$ (i.e., the set of indices $i \in N$ such that $r_i^{\mathcal{C}} \neq 0$) is the set of agents forming coalition $\mathcal{C}$. For all $i \in N$ and all coalition $\mathcal{C} \in S$, $r_i^{\mathcal{C}} \in [0,1]^n$ represents the fraction of resource that agent i contributes to coalition $\mathcal{C}$; hence, $\sum_{\mathcal{C}\in S} r_i^{\mathcal{C}} \leq 1$ (i.e., agent i cannot contributes more than 100% of its resource). A payoff distribution for a CS S of size k is defined by a k-tuple $x = (x^1, \ldots, x^k)$ where $x^{\mathcal{C}}$ is the payoff distribution that the agents obtain for coalition $\mathcal{C}$. If an agent is not in the coalition, it must not receive any payoff for this coalition, hence $(r_i^{\mathcal{C}} = 0) \Rightarrow (x_i^{\mathcal{C}} = 0)$. The total payoff of agent i is the sum of its payoffs over all coalitions $p_i(CS, x) = \sum_{\mathcal{C}=1}^{k} x_i^{\mathcal{C}}$. The efficiency criterion becomes $\forall r^{\mathcal{C}} \in S$, $\sum_{i\in N} x_i^{\mathcal{C}} = v(r^{\mathcal{C}})$. An imputation is an efficient payoff distribution that is also individually rational. We denote by $I(S)$ the set of all imputations for the CS S.

---

[1] An earlier version is [18]

We are now ready to define the overlapping core. One issue is the kind of permissible deviations: when an agent deviates, she can completely leave some coalitions, reduce her contribution in other coalitions, or contributes to new coalitions. If she stills contribute to a coalition containing non-deviating agents, how should they behave? They first may refuse to give any payoff to the deviating agent, as she is seen as not trustworthy. Agents that are not affected by the deviation may, however, consider that the deviators agents did not fail them, and consequently, they may continue to share payoffs with the deviators. A last case occurs when the deviators are decreasing their implication in a coalition. This coalition may no longer perform the same tasks, but it can still perform some. If there is enough value to maintain the payoff of the non-deviators, the deviators may be allowed to share the surplus generated. Each of these behaviors give raise to different types of deviations, and consequently, different definition of a core: the conservative core, the refined core and the optimistic core. The paper also provides a characterization of conservative core, properties of the different core, including a result showing that convex overlapping coalitional games have a non-empty core.

12.9 Trust

The notion of trust can be an important metric to determine whom to interact with. This is particularly important when the coalition is expected to live for a long term. In [7], an agent computes a probability of success of a coalition, based on a notion of trust which can be used to eliminate some agents from future consideration. This probability is used to estimate the value of different coalitions and help the agent in deciding which coalition to join or form. In [43], the decision to leave or join a coalition is function of the trust put in other agents. In this paper, the concept of trust is defined as a belief that agents will have successful interaction in the future; hence, trust is used to consider a subset of the entire population of agents for the formation of future coalitions. Trust is used to compute coalitions, but agents do not compute a payoff distribution. Another work that emphasises trust is [22] which introduces the concept of clans. A clan is formed by agents that trust each other with long-term commitments. Given the trust and an estimate of local gain, agents can accept to join a clan. The idea behind this work is that agents that trust each other will be collaborative. Moreover, when an agent needs to form a coalition of agents, it will only search partners in the clan, which reduces the search space. Trust can therefore be very effective for scaling up in large society of agents.

12.10 Learning

When agents have to repeatedly form coalitions in the presence of the same set of agents, learning can be used to improve performance of the coalition formation process both in terms of speed of the process and in terms of better valuation.

A basic model of iteratively playing many coalitional games is presented in [32]: at each time step, a task is offered to agents that are already organised into coalitions. The task is awarded to the best coalition. The model is made richer in [33] where the agents can estimate the value of a coalition and have a richer set of actions: as the agents can fire members from a coalition, join a different coalition, or leave a coalition to replace some agents in a different coalition. However, in both works, the agents are not learning, they have a set of static strategies. Empirical experiments compare the results over populations using either the same strategy or a mix of strategies.

Chalkiadakis and Boutilier also consider a repeated coalition formation problem [14, 15, 16]. The setting is a task allocation problem where agents know their own types (i.e., skill to perform some type of tasks), but do not know the ones of other agents in the population. Each time a coalition is formed, the agents will receive a value for that coalition. From the observation of this value, the agents can update a belief about the types of other agents. When an agent is reasoning about which coalition to form, it uses its beliefs to estimate the value of the coalition. This problem can be formulated using a POMPD (Partially observable Markov Decision Process) where the agents are maximising the long-term value of their decision over the repetition of the coalition formation process. Solving a POMPD is a difficult task, and the POMPD for the coalition formation problem grows exponentially with the number of agents. In [14], a myopic approach is proposed. More recently, Chalkiadakis and Boutilier propose additional algorithms to solve that POMPD, and empirically compare the solutions [15].

Bibliography

[1] Sherief Abdallah and Victor Lesser. Organization-based cooperative coalition formation. In *IAT 04*, 2004.

[2] Stéphane Airiau and Sandip Sen. A fair payoff distribution for myopic rational agents. In *Proceedings of the Eighth International Conference on Autonomous Agents and Multiagent Systems (AAMAS-09)*, May 2009.

[3] Stéphane Airiau and Sandip Sen. On the stability of an optimal coalition structure. In *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI-2010)*, pages 203–208, August 2010.

[4] J.-P. Aubin. *Mathematical Methods of Game and Economic Theory*. North-Holland, 1979.

[5] María-Victoria Belmonte, Ricardo Conejo, José-Luis Pérez de-la Cruz, and Francisco Triguero Ruiz. A stable and feasible payoff division for coalition formation in a class of task oriented domains. In *ATAL '01: Revised Papers from the 8th International Workshop on Intelligent Agents VIII*, pages 324–334, London, UK, 2002. Springer-Verlag.

[6] María-Victoria Belmonte, Ricardo Conejo, José-Luis Pérez de-la Cruz, and Francisco Triguero Ruiz. A robust deception-free coalition formation model. In *Proceedings of the 2004 ACM symposium on Applied computing (SAC '04)*, pages 469–473, New York, NY, USA, 2004. ACM Press.

[7] Bastian Blankenburg, Rajdeep K. Dash, Sarvapali D. Ramchurn, Matthias Klusch, and Nicholas R. Jennings. Trusted kernel-based coalition formation. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 989–996, New York, NY, USA, 2005. ACM Press.

[8] Bastian Blankenburg, Minghua He, Matthias Klusch, and Nicholas R. Jennings. Risk-bounded formation of fuzzy coalitions among service agents. In *Proceedings of 10th International Workshop on Cooperative Information Agents*, 2006.

[9] Bastian Blankenburg and Matthias Klusch. On safe kernel stable coalition forming among agents. In *Proceedings of the third International Joint Conference on Autonomous Agents and Multiagent Systems*, 2004.

[10] Bastian Blankenburg and Matthias Klusch. Bsca-f: Efficient fuzzy valued stable coalition forming among agents. In *Proceedings of the IEEE International Conference on Intelligent Agent Technology (IAT)*. IEEE Computer Society Press, 2005.

[11] Bastian Blankenburg, Matthias Klusch, and Onn Shehory. Fuzzy kernel-stable coalitions between rational agents. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems (AAMAS-03)*. ACM Press, 2003.

[12] Christopher H. Brooks and Edmund H. Durfee. Congregation formation in multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 7(1-2):145–170, 2003.

[13] Philippe Caillou, Samir Aknine, and Suzanne Pinson. A multi-agent method for forming and dynamic restructuring of pareto optimal coalitions. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 1074–1081, New York, NY, USA, 2002. ACM Press.

[14] Georgios Chalkiadakis and Craig Boutilier. Bayesian reinforcement learning for coalition formation under uncertainty. In *Proceedings of the third International Joint Conference on Autonomous Agents and Multiagent Systems(AAMAS'04)*, 2004.

[15] Georgios Chalkiadakis and Craig Boutilier. Sequential decision making in repeated coalition formation under uncertainty. In *Proc. of the 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS-08)*, May 2008.

[16] Georgios Chalkiadakis and Craig Boutilier. Sequentially optimal repeated coalition formation. *Autonomous Agents and Multi-Agent Systems*, to appear, 2010. Published online November 2010.

[17] Georgios Chalkiadakis, Edith Elkind, and Nicholas R. Jennings. Simple coalitional games with beliefs. In *Proceedings of the 21st international jont conference on Artifical intelligence*, pages 85–90, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.

[18] Georgios Chalkiadakis, Edith Elkind, Evangelos Markakis, and Nicholas R. Jennings. Overlapping coalition formation. In *Proceeedings of the 4th International Workshop on Internet and Network Economics (WINE2008)*, pages 307–321, 2008.

[19] Georgios Chalkiadakis, Edith Elkind, Evangelos Markakis, and Nicholas R. Jennings. Cooperative games with overlapping coalitions. *Journal of Artificial Intelligence Research*, 39:179–216, 2010.

[20] Vincent Conitzer and Tuomas Sandholm. Computing shapley values, manipulating value division schemes, and checking core membership in multi-issue domains. In *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI-04)*, pages 219–225, 2004.

[21] Viet Dung Dang, Rajdeep K. Dash, Alex Rogers, and Nicholas R. Jennings. Overlapping coalition formation for efficient data fusion in multi-sensor networks. In *Proceedings of the Twenty-First Conference on Artificial Intelligence (AAAI-06)*, pages 635–640, 2007.

[22] Nathan Griffiths and Michael Luck. Coalition formation through motivation and trust. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems (AAMAS03)*, New York, NY, USA, 2003. ACM Press.

[23] Bryan Horling and Victor Lesser. A survey of multi-agent organizational paradigms. *The Knowledge Engineering Review*, 19:281–316, 2004.

[24] Samuel Ieong and Yoav Shoham. Bayesian coalitional games. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence (AAAI-08)*, pages 95–100, 2008.

[25] Steven P. Ketchpel. Forming coalitions in the face of uncertain rewards. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 414–419, August 1994.

[26] Matthias Klusch and Onn Shehory. Coalition formation among rational information agents. In Rudy van Hoe, editor, *Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, Eindhoven, The Netherlands, 1996.

[27] Matthias Klusch and Onn Shehory. A polynomial kernel-oriented coalition algorithm for rational information agents. In *Proceedings of the Second International Conference on Multi-Agent Systems*, pages 157 – 164. AAAI Press, December 1996.

[28] Sarit Kraus, Onn Shehory, and Gilad Taase. Coalition formation with uncertain heterogeneous information. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 1–8. ACM Press, 2003.

[29] Hoong Chuin Lau and Lei Zhang. Task allocation via multi-agent coalition formation: Taxonomy, algorithms and complexity. In *15th IEEE International*

Conference on Tools with Artificial Intelligence (ICTAI 2003), pages 346–350, November 2003.

[30] Katia Lerman and Onn Shehory. Coalition formation for large-scale electronic markets. In *Proceedings of the Fourth International Conference on MultiAgent Systems (ICMAS-2000)*. IEEE Computer Society, July 2000.

[31] M. Mares. Fuzzy cooperative games: cooperation with vague expectations. *Studies in fuzziness and soft computing*, 72, 2001.

[32] Carlos Mérida-Campos and Steven Willmott. Modelling coalition formation over time for iterative coalition games. In *Proceedings of the third International Joint Conference on Autonomous Agents and Multiagent Systems(AAMAS'04)*, 2004.

[33] Carlos Mérida-Campos and Steven Willmott. The effect of heterogeneity on coalition formation in iterated request for proposal scenarios. In *Proceedings of the Forth European Workshop on Multi-Agent Systems (EUMAS 06)*, 2006.

[34] I. Nishizaki and M. Sakawa. Masatoshi. Fuzzy and multiobjective games for conflict resolution. *Studies in Fuzziness and Soft Computing*, 64, 2001.

[35] Naoki Ohta, Vincent Conitzer, Yasufumi Satoh, Atsushi Iwasaki, and Makoto Yokoo. Anonimity-proof shapley value: Extending shapley value for coalitional games in open environments. In *Proc. of the 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS-09)*, May 2009.

[36] Michal Pěchouček, Vladimír Mařík, and Jaroslav Bárta. A knowledge-based approach to coalition formation. *IEEE Intelligent Systems*, 17(3):17–25, 2002.

[37] Ariel D. Procaccia and Jeffrey S. Rosenschein. The communication complexity of coalition formation among autonomous agents. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems AAMAS-06*, May 2006.

[38] Tuomas Sandholm and Victor R. Lesser. Coalitions among computationally bounded agents. *AI Journal*, 94(1–2):99–137, 1997.

[39] Onn Shehory and Sarit Kraus. Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101(1-2):165–200, May 1998.

[40] Onn Shehory and Sarit Kraus. Feasible formation of coalitions among autonomous agents in nonsuperadditve environments. *Computational Intelligence*, 15:218–251, 1999.

[41] Leen-Kiat Soh and Costas Tsatsoulis. Satisficing coalition formation among agents. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 1062–1063, New York, NY, USA, 2002. ACM Press.

[42] Fernando Tohmé and Tuomas Sandholm. Coalition formation processes with belief revision among bounded-rational self-interested agents. *Journal of Logic and Computation*, 9(6):793–815, 1999.

[43] Julita Vassileva, Silvia Breban, and Michael Horsch. Agent reasoning mechanism for making long-term coalitions based on decision making and trust. *Computational Intelligence*, 18(4):583–595, 2002.

[44] Makoto Yokoo, Vincent Conitzer, Tuomas Sandholm, Naoki Ohta, and Atsushi Iwasaki. Coalitional games in open anonymous environments. In *Proceedings of the Twentieth National Conference on Artificial Intelligence*, pages 509–515. AAAI Press AAAI Press / The MIT Press, 2005.

[45] Makoto Yokoo, Vincent Conitzer, Tuomas Sandholm, Naoki Ohta, and Atsushi Iwasaki. A compact representation scheme for coalitional games in open anonymous environments. In *Proceedings of the Twenty First National Conference on Artificial Intelligence*, pages –. AAAI Press AAAI Press / The MIT Press, 2006.