Reducing the Search Space of Coalition Structures based on Smart Grids' Domain Information

Gabriel de Oliveira Ramos[1] and Ana L. C. Bazzan[1]

[1] Instituto de Informática, Universidade Federal do Rio Grande do Sul,
P. O. Box 15064, Porto Alegre, RS, Brazil
{goramos,bazzan}@inf.ufrgs.br

Abstract. Smart grids have received great attention in recent years. Among many technologies that smart grids are made of, the concept of vehicle-to-grid (V2G) arises, which allows using electric vehicles' batteries to provide energy back to the grid when needed. An interesting research approach is the formation of coalitions among electric vehicles to participate in V2G sessions in a cost effective way. In this paper, we propose a technique that uses smart grids' domain information to establish constraints on coalition formation process, promoting the pruning of the coalition structure's search space. We evaluate our approach by showing that it can drastically prune the search space, which makes possible forming coalitions for a relatively higher number of agents.

1 Introduction

Electric power is one of the key factors to the modern economies' progress. In most countries, electric power generation strongly depends on non-renewable, highly polluting energy sources whose availability is becoming increasingly scarce. In order to reduce impacts on the environment, many governments around the world have been focusing on the transition to a low carbon economy. One of the major challenges arising from this transition refers to the modernization of the energy infrastructure (the grid).

According to the U. S. Department of Energy [1], in most countries the grid is precarious and inefficient, endowed with little or no redundancy and responsible for severe impacts on the environment. In such scenario the concept of smart grids emerges, which is described as "a fully automated power delivery network that monitors and controls every customer and node, ensuring a two-way flow of electricity and information between the power plant and the appliance, and all points in between" [1].

The concept of smart grids is closely related to computing, since it was conceived under a solid foundation from engineering, communications, distributed intelligence, automation and information exchange [2, 1]. Smart grids also represent a large field of research for multiagent systems, given its distributed nature.

In addition to the concept of smart grids, the concept of vehicle-to-grid (V2G) has been used. During V2G sessions, electric vehicles (EVs) can sell part of the

energy stored in their batteries to the grid. This occurs when production is not able to meet demand [3]. However, participating in V2G sessions in a cost effective way is not a trivial task, given the insufficient energy capacity and availability of EVs. In order to address such difficulty, many researchers have proposed the formation of coalitions among EVs, as in [4–7]. Such approach has shown to be an efficient way to increase the EVs' profitability [8].

Coalition formation is a research topic that has received great attention in the field of multiagent systems. According to [9], a coalition can be defined as a group of agents that decide to cooperate in order to achieve a common goal. However, forming coalitions is not just grouping agents, but grouping them in order to obtain the greatest possible reward, which has been proven to be NP-complete [10]. Coalition formation includes three activities [10]:

1. Coalition structure generation: partitioning the set of agents into disjoint and exhaustive coalitions. This partition is called a coalition structure (CS).
2. Solving the optimization problem of each coalition: the agents must coordinate themselves in order to accomplish the tasks and to use the available resources, increasing the value obtained by the coalitions they are in.
3. Dividing the obtained value among the agents.

In [11], Rahwan et al. proposed a generic approach for finding near-optimal coalition structures by reducing the search space. However, a greater reduction may be achieved by using specific domain information. In this paper we focus on the reduction of the coalition structures' search space based on smart grids' domain information. More specifically, we investigate how coalitions can be formed among EVs, drawing upon physical constraints of the grid, in order to reduce the complexity of the process. Thus, we propose a technique capable of identifying agents that cannot form coalitions among themselves due to specific constraints. On identifying such a set of agents, we can prune the search space by removing all unfeasible coalition structures.

This paper is organized as follows. Section 2 presents relevant related work. In Section 3 the background of coalition structure generation is presented. In Section 4 the problem is modeled. In Section 5 we detail the proposed technique. Section 6 evaluates our proposal. Finally, Section 7 presents the conclusions and the future research directions.

2 Related Work

In this section, we review relevant work in the field of multiagent systems and smart grids. We begin with the work of Chalkiadakis et al. [4], which addresses coalition formation between distributed energy resources (DERs) to form virtual power plants (VPPs). DERs are intermittent renewable energy generators with small-to-medium energy capacity, like wind turbines, solar panels etc. Their approach suggests grouping DERs to promote reliability and efficiency on production predictions. To incentivize DERs to provide accurate estimates, a mechanism rewards good estimates and punishes the bad ones. However, this approach

has a greater focus on mechanism design than in coalition formation, disregarding how far the solution is from the optimum.

In the work of Vasirani et al. [5], the focus is on coalition formation among wind turbines and EVs. The goal on forming such coalitions is to increase the reliability of these intermittent energy sources through the use of EVs' batteries. Based on that, wind turbines could improve their profits, whereas EVs could make a profit by renting space in their batteries to store energy. Notwithstanding, aspects concerning the optimization of the coalition structure are not taken into account in their proposal.

Kamboj et al.'s approach [6] proposes forming coalitions of EVs to act in the regulation market. The goal of the regulation market is to increase the grid stability by ensuring energy availability. The regulation market basically provides power to the grid, whenever demand exceeds supply, and store energy, whenever supply exceeds demand, through batteries (readily store and supply; expensive) and generators (generate energy; delayed start, polluting). Thereby, considering EVs remain parked most of the time [3], the proposal is using EVs' batteries to reduce costs and improve efficiency of the regulation market. More recently, this approach was extended and deployed in real world for five EVs [7]. However, in both cases coalition formation is made in an ad hoc fashion, without concerning with the optimal solution.

Therefore, it is clear that existing works have given greater importance to applications of smart grids than to coalition formation itself. Consequently, it becomes evident that this is a major research challenge that requires new ideas to evolve and to bring new contributions to this field.

3 Background of Coalition Structure Generation

In this section we briefly present the background of coalition structure generation. Coalition formation is commonly studied as characteristic function games (CFGs) [9, 11, 10], where the value of each coalition C is given by $v(C)$ [12]. The value v represents how beneficial the formation of such a coalition could be.

The coalition structures' search space grows extremely fast: given a set of a agents, the number of possible coalitions is $2^a - 1$ and of coalition structures is $\mathrm{O}(a^a)$ and $\omega(a^{\frac{a}{2}})$ [10]. This search space can be represented by a graph, where coalition structures are grouped in levels according to the number of coalitions they have, as shown in Fig. 1. The number of coalition structures of a level l can be obtained by the Stirling number of the second kind [10], given by Eq. (1), where $Z(a, a) = Z(a, 1) = 1$.

$$Z(a, l) = lZ(a - 1, l) + Z(a - 1, l - 1) \ . \tag{1}$$

The exact number of coalition structures can be obtained by summing up the resulting Stirling number of all levels, as in Eq. (2).

$$\sum_{l=1}^{a} Z(a, l) \ . \tag{2}$$

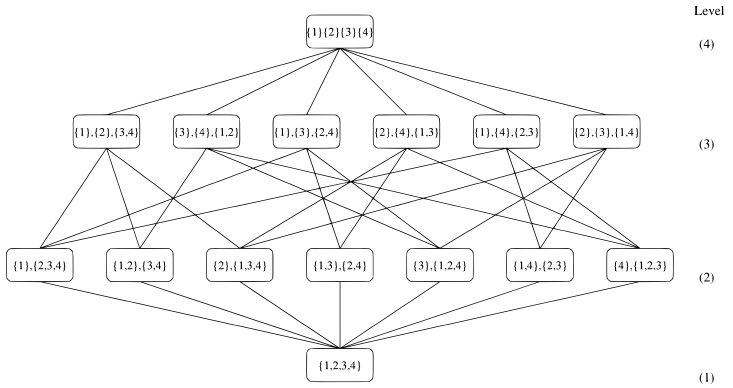


Fig. 1. Coalition structure graph for four agents (extracted from [10])

Finding the optimal coalition structure can be seen as searching in the coalition structure graph, which is unfeasible due to computational complexity. Sandholm et al. [10] proved that searching the lowest two levels of the graph is sufficient to establish a worst case bound on the quality of the coalition structure. Additionally, they showed that by searching further it is possible to establish a progressively lower tight bound. Since then, their contribution has been the cornerstone in which all recent works have been based on.

In [11], Rahwan et al. proposed an anytime algorithm to find near-optimal coalition structures. To understand how it works, some basic definitions are necessary. The set of agents is represented by A, with $a = |A|$. A coalition is a subset $C \subseteq A$, where $v(C)$ represents its value. A coalition structure $CS \in \mathcal{CS}$ is a partition of A into disjoint and exhaustive coalitions, whose value $V(CS) = \sum_{C \in CS} v(C)$. The goal of the algorithm is to find the optimal coalition structure $CS^* = \arg\max_{CS \in \mathcal{CS}} V(CS)$, i.e., the one with the greatest value.

The proposed algorithm in [11] is based on a novel representation of the search space, where coalition structures are grouped by the size of coalitions they have (called configurations). For example, both coalition structures $\{\{1\}, \{2, 3\}\}$ and $\{\{3\}, \{1, 2\}\}$ follow the configuration $\{1, 2\}$. So, let $CL_S \in \mathcal{CL}$ be the set of all coalitions of size $s \in \{1, 2, ..., a\}$ and let $G_1, G_2, ...G_{|\mathcal{G}_{\mathcal{CS}}|} \in \mathcal{G}_{\mathcal{CS}}$ be the set of all possible configurations.

Based on those definitions, the algorithm is divided into three stages:

1. Pre-processing: performs the search on every $CL_S \in \mathcal{CL}$, obtaining the maximum and average $v(C)$ of each list.
2. Choosing the optimal configuration: uses the values obtained in the previous stage to select the element $G \in \mathcal{G}_{\mathcal{CS}}$ that is most likely to have the CS^*. This stage identifies only one configuration, which allows performing a search in only a small portion of the graph in the next stage.
3. Finding the CS^*: where all $V(CS)$ of the selected G are computed to find the optimal solution CS^*. This is the most computationally costly stage.

4 Problem Modeling

The scenario presented in our work is a smart grid where EVs sell their surplus energy on V2G sessions. In this scenario, the grid incentivizes these EVs to form coalitions by offering more attractive prices when energy supply is greater.

The aim of the EV agents is to maximize their profits. Each EV has an energetic availability, i.e., the amount of surplus energy that can be sold. Coalitions are formed at the beginning of a time period $t \in \mathcal{T}$ and lasts until the end of that period. As in [4], each time t corresponds to a half-hour period.

As a common practice in literature [9, 11, 10], this problem is modeled by means of a characteristic function game (CFG). The characteristic function here represents how much a coalition should receive beyond the normal price. For example, if the normal price were \$1.00 per kWh and for a given coalition C, with five agents, the grid would pay \$1.05 per kWh, then $v(C)$ would be 0.05.

5 Domain Information

In this section we describe our technique to prune the search space based on domain information. To this end, an important issue must be addressed: the energy provided by EVs must supply only consumers who are located close to them. Otherwise, if the supplied energy travels long distances until reaching the consumer, the grid could be overloaded. The overload may occur because the distribution over long distances has a higher chance of overlapping with local energy distribution [1, 7]. Therefore, distance can be used as an infrastructure constraint on coalition formation.

The distance constraint is given by the constant α, which is the maximum distance that must exist among agents of the same coalition. Given this, let $D^a \in \mathcal{D}$ be the set of constraints of agent a, i.e., the agents with whom it cannot form coalitions. Based on the agents' constraints, it is possible to find a subset D in which all agents have constraints among themselves. However, different D sets may exist, depending on agents' point of view. For example, for a set of agents $A = \{a1, a2, a3, a4\}$, if the distance constraint would prevent agents $a1$ and $a2$ to form a coalition and also prevent agents $a3$ and $a4$ to do so, then two different D sets would exist: $D_1 = \{a1, a2\}$ and $D_2 = \{a3, a4\}$. So, let $\mathbb{D} = \{D_1, D_2, ..., D_{|\mathbb{D}|}\}$ be the set of all possible D sets.

Our technique works by selecting the $D^* = \arg\max_{D \in \mathbb{D}} |D|$ set and then pruning the lowest $|D^*| - 1$ levels of the coalition structure graph. This approach is valid because, since lower levels have the coalition structures with less partitions than those in the higher ones, it is clear that all coalition structures in the lowest $|D^*| - 1$ levels ignore at least one of the D^* constraints. For example, suppose that $A = \{a1, a2, a3, a4\}$ and $D^* = \{a1, a2, a3\}$. In order to keep the D^*'s agents separated, a coalition structure must have at least three coalitions. Thus, the two lowest levels of the graph can be pruned, since all coalition structures in these levels have two or less coalitions, as can be seen in Fig. 1.

Thus, we need to find the set D^* to find out how many levels of the search space can be pruned. Such a task is performed by Alg. 1. This algorithm requires

the set of agents A, the constant α and the distance among the agents. So, let d_b^a be the distance between the agents a and $b \in A$. Initially, the algorithm identifies the D^a set of each agent (shown at line 4 of Alg. 1) and then identifies the greatest D^a for optimization purposes (line 5 of Alg. 1).

Algorithm 1: Finding how many levels can be pruned based on D^*

```
   Requires: A: the set of agents        α: the distance constraint
             : d_b^a for all a ∈ A and b ∈ A/a: the distance among each agents
   Returns : the size of D*
 1 function findBestD ()
 2 begin
 3     initialize s ← 0, g ← 0;
       /* computes the set of constraints D^a of each agent a                    */
 4     let D^a ∈ 𝒟 ← arg_{a∈A,b∈A/a} d_b^a > α;
 5     let g ← max_{D^a∈𝒟} |D^a|
       /* iterates over all D^a to identify the size of D*                       */
 6     foreach D^a ∈ 𝒟 do
 7         foreach b ∈ D^a do
 8             if |D^a| == 1 then  s_D ← 2;
 9             else
10                 s_D ←findDs(D^a)+1;
11                 D^a ← D^a/b;
12             end if
13             if s_D > g then  return s_D;
14             else if s_D > s then  s ← s_D;
15         end foreach
16     end foreach
17     return s;
18 end findBestD;
19
   /* recursively iterates over a given D^a to identify the agents that have constraints
      among themselves                                                            */
20 function findDs(D^a)
21 begin
22     let b ← first element of D^a;
23     let D^b ← arg_{c∈D^a/b,c>b} d_c^b > α;
24     if |D^b| = 1 then  return 2;
25     else if |D^b| > 1 then  return findDs(D^b)+1;
26     else  return 1;
27 end findDs;
28
```

Next, for each agent b of every $D^a \in \mathcal{D}$, the algorithm finds the greatest D^b possible, which contains all agents of D^a that have constraints with the agent b (recursive function of lines 16 to 23 of Alg. 1). To avoid searching on redundant D^a sets, when a D^b set is being created the algorithm considers only agents c whose index is greater than that of b (line 19), assuming that the agents are ordered by their indexes. This process ends only after all $D^a \in \mathcal{D}$ were analyzed or when a D^b set whose size equals to g has been encountered. The stopping criteria defined by g (line 12) interrupts the search if all agents of the greatest D^a have constraints among themselves, meaning that this is the D^*.

The following example presets a simple scenario for six agents to help understanding how this technique works.

Example 1. Let $A = \{a1, a2, a3, a4, a5, a6\}$, $\alpha = 100$m and the geographical distribution of Fig. 2.

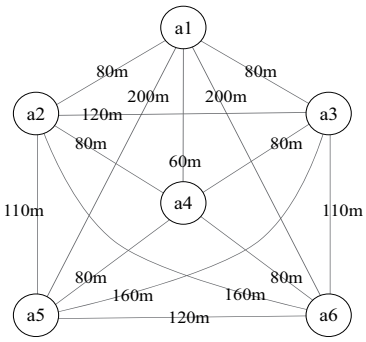


Fig. 2. Hypothetical scenario of example 1

In this example, the agent $a1$ cannot form coalitions with agents $a5$ and $a6$ due to the distance constraint, meaning that $D^{a1} = \{a5, a6\}$. Similarly, the agents $a2$, $a3$, $a5$ and $a6$ cannot form coalitions among themselves, i.e., $D^{a2} = \{a3, a5, a6\}$, $D^{a3} = \{a2, a5, a6\}$, $D^{a5} = \{a2, a3, a6\}$ and $D^{a6} = \{a2, a3, a5\}$. Finally, agent $a4$ has no constraints with the others, i.e., $D^{a4} = \emptyset$. Therefore, we have $\mathcal{D} = \{\{a5, a6\}, \{a3, a5, a6\}, \{a2, a5, a6\}, \emptyset, \{a2, a3, a6\}, \{a2, a3, a5\}\}$. Now the algorithm iterates over every $D^a \in \mathcal{D}$ to find the D^*. In the first iteration, D^{a1} is analyzed and results in $D_1 = \{a1, a5, a6\}$. In the second iteration, while analyzing D^{a2}, the result is $D_2 = \{a2, a3, a5, a6\}$. No additional sets are found through the remaining iterations (because $a3$, $a5$ and $a6$ have basically the same constraints as $a2$, and because $a4$ has no constraints). Therefore, as the size of D_2 is greater than that of D_1, the result is $D^* = D_2 = \{a2, a3, a5, a6\}$, meaning that the three lowest levels of the coalition structure graph can be pruned. This prune can be done because, to keep these agents in separate coalitions, at least four partitions on coalition structures are required. This happens only in the three highest levels of the graph. Based on Eq. (2), there are 203 possible coalition structures for six agents. Similarly, the size of the three lowest levels of the graph can be calculated by Eq. (1), resulting in 122 coalition structures. Therefore, this pruning corresponds to 60.09% of the search space.

Although this technique can make a relatively large pruning in the graph, it does not always produce a good result. In scenarios where most agents are closely located and within the distance α, i.e., there are too few constraints, the D^* set would be very small. This would reduce the effectiveness of the technique.

6 Evaluation

In this section we present the evaluation of our technique. We begin by comparing our technique to the Rahwan et al's one [11]. In their work, results were presented for 20 agents at most. For the sake of comparison, we show results for 10, 15 and 20 agents, varying the size of the D^* set, to compare how much the search

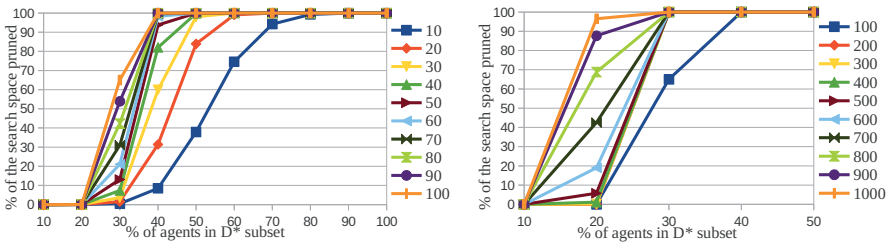
Table 1. Pruned percentage performed by our mechanism compared to [11]

a	$\lvert\mathcal{CS}\rvert$	$\lvert D^*\rvert$	Pruned CSs	% pruned	% pruned by [11]
10	115975	4	9842	8.48631170510886	< 1
10	115975	6	86472	74.5608967449881	< 1
10	115975	8	115179	99.3136451821513	< 1
15	1382958545	6	255514355	18.4759229352027	≈ 40
15	1382958545	9	1301576801	94.1153880357274	≈ 40
15	1382958545	12	1382847419	99.9919646181441	≈ 40
20	51724158235372	8	16244652278171	31.4063154092317	≈ 50
20	51724158235372	12	51245294979716	99.0741980691558	≈ 50
20	51724158235372	16	51724135127267	99.9999553243479	≈ 50

space was pruned. As shown in Table 1, both approaches perform better as the number of agents increases. However, for a small number of agents our technique performs better than Rahwan et al's approach. Moreover, it can run for a large number of agents, while [11] runs for up to 20 agents.

We now show results for a greater number of agents. In Fig. 3(a), results were generated for samples of 10 to 100 agents (only multiples of ten). For each sample, the graph shows how much space may be pruned according to the percentage of agents in the D^* set (also for multiples of ten). Figure 3(b) shows the same as in Fig. 3(a), however for 100 to 1000 agents (multiples of hundred).

As shown in Fig. 3, the percentage of the search space to be pruned increases very fast when more than 20% of the agents are in the D^* set. This growth is intensified as the number of agents increase. Also, the pruned percentage approaches, but never actually gets at, 100%. This happens earlier as the number of agents increase. Hence, a trend on the curves behavior can be observed, namely that the pruned percentage drastically increases when more than 20% of the agents are in the D^* set, and rapidly tends to 100%, when more than 30% of the agents are in the D^* set. However, although the search space can be drastically reduced based on our mechanism, this does not mean that the remaining search



(a) 10 to 100 agents (b) 100 to 1000 agents

Fig. 3. Pruned percentage of the search space for different sizes of D^*

EV	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	-																			
2	100	-																		
3	102	200	-																	
4	146	242	45	-																
5	224	124	321	360	-															
6	322	222	420	460	100	-														
7	90	128	122	149	226	324	-													
8	313	215	406	443	94	54	300	-												
9	211	297	120	81	406	505	181	480	-											
10	178	260	98	70	367	465	141	440	40	-										
11	112	128	150	175	212	308	28	281	200	160	-									
12	202	122	285	318	86	166	171	132	350	310	150	-								
13	357	260	449	485	139	73	341	45	520	480	320	170	-							
14	140	164	156	172	242	335	50	304	182	143	36	173	341	-						
15	284	208	358	386	140	172	238	120	406	367	212	86	139	224	-					
16	206	255	171	162	335	427	128	393	128	100	128	262	428	94	300	-				
17	233	200	276	294	208	278	156	233	297	260	128	122	260	122	124	181	-			
18	297	233	361	385	181	216	239	164	396	357	211	114	178	215	45	282	100	-		
19	378	297	452	479	206	191	331	140	495	456	305	177	126	313	94	382	201	100	-	
20	260	305	220	204	377	465	180	427	153	136	177	300	459	141	326	54	202	301	400	-

Fig. 4. Distance (in meters) among the agents of the hypothetical scenario

space is small: for a thousand agents, given $|D^*| = 30\%$, the search space would be reduced from the order of 10^{1927} to 10^{1857}, which obviously does not help.

Now we present a hypothetical smart grid scenario to illustrate how our technique works. We use a small neighborhood where 20 EVs are parked and plugged into the grid, ready to sell their surplus energy through V2G sessions. The distance among the agents is depicted in Fig. 4.

In this hypothetical scenario, the EVs' energy availability were uniformly selected from 0.5 to 2 kWh. Distance constraint is $\alpha = 100$m. A unit of energy corresponds to 1 kWh and costs \$0.50. The coalition values were generated based on the coalitions' size and amount of energy. Based on these information, the algorithm was able to identify $|D^*| = 11$, i.e., 10 levels of the graph may be pruned.

In this scenario, for 20 agents there are $\approx 5 \times 10^{13}$ coalition structures. After the pruning, only 4.6% has remained. So, the algorithm found $CS^* = \{\{1\}, \{2\}, \{17\}, \{20\}, \{3, 4\}, \{7, 11\}, \{8, 13\}, \{9, 10\}, \{14, 16\}, \{5, 6, 12\}, \{15, 18, 19\}\}$, whose configuration is $\{1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3\}$ and value is $V(CS^*) = 0.216$.

7 Conclusions

In this paper we presented an algorithm to reduce the coalition structures' search space. The algorithm focuses on identifying agents that cannot form coalitions among themselves, based on domain information, to perform a considerable prune on the search space. This technique may be applied in smart grids to comply the distance constraint existing among EVs. Our approach shown to be efficient, performing a good pruning in the search space.

For future work we expect to improve our technique to prune not only entire levels, but also specific portions within a given level. To achieve this we may build on the representation proposed in [11], pruning coalition lists and configurations. Also, we may concentrate on how the search space could be represented in order to allow new pruning techniques.

Another approach under investigation is the idea of more valuable players [9], i.e., agents that aggregate additional value to the coalitions they join. Based on this concept, it is possible to identify the exact level where the optimal solution is. Here we expect to identify elements of smart grids that could aggregate additional value like buildings, whose energy capacity is supposed to be greater than the EVs. Buildings may represent an attractive energy source, given that they may have their own generators and even a lot of EVs parked in.

Acknowledgements

We are very grateful to Anderson Tavares, Cássio Oliveira and the anonymous reviewers for their valuable comments. The authors are partially supported by CNPq and FAPERGS.

References

1. U. S. Department of Energy: Grid 2030: A national vision for electricity's second 100 years (2003)
2. Ramchurn, S., Vytelingum, P., Rogers, A., Jennings, N.: Putting the "smarts" into the smart grid: A grand challenge for artificial intelligence. Communications of the ACM **55**(4) (2012) 86–97
3. Kempton, W., Tomić, J.: Vehicle-to-grid power implementation: From stabilizing the grid to supporting large-scale renewable energy. Journal of Power Sources **144**(1) (2005) 280–294
4. Chalkiadakis, G., Robu, V., Kota, R., Rogers, A., Jennings, N.R.: Cooperatives of distributed energy resources for efficient virtual power plants. In: Proceedings of 10th International Conference on Autonomous Agents and Multiagent Systems, Taipei, Taiwan (2011) 787–794
5. Vasirani, M., Kota, R., Cavalcante, R., Ossowski, S., Jennings, N.: Using coalitions of wind generators and electric vehicles for effective energy market participation. In: Proceedings of 10th International Conference on Autonomous Agents and Multiagent Systems. (2011) 1099–1100
6. Kamboj, S., Decker, K., Trnka, K., Pearre, N., Kern, C., Kempton, W.: Exploring the formation of electric vehicle coalitions for vehicle-to-grid power regulation. In: First International Workshop on Agent Technology for Energy Systems. (2010) 1–8
7. Kamboj, S., Kempton, W., Decker, K.S.: Deploying power grid-integrated electric vehicles as a multi-agent system. In: Proceedings of 10th International Conference on Autonomous Agents and Multiagent Systems, Taipei, Taiwan (2011) 13–20
8. Pudjianto, D., Ramsay, C., Strbac, G.: Virtual power plant and system integration of distributed energy resources. Renewable Power Generation, IET **1**(1) (2007) 10–16
9. Bazzan, A.L.C., Dahmen, S.R.: Targeted search in coalition structures. In: Proceedings of the 5th European Workshop on Multi-agent Systems, Hammamet, Tunisia (2007) 180–192
10. Sandholm, T., Larson, K., Andersson, M., Shehory, O., Tohmé, F.: Coalition structure generation with worst case guarantees. Artificial Intelligence **111**(1–2) (1999) 209–238

11. Rahwan, T., Ramchurn, S.D., Dang, V.D., Jennings, N.R.: Near-optimal anytime coalition structure generation. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence. (2007) 2365–2371
12. Schmeidler, D.: The nucleolus of a characteristic function game. SIAM Journal on applied mathematics **17**(6) (1969) 1163–1170