

Reduction of Coalition Structure’s Search Space based on Domain Information: an Application in Smart Grids

Gabriel de Oliveira Ramos¹ and Ana L. C. Bazzan¹

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 are used in smart grids, the concept of vehicle-to-grid (V2G) arises, which allows the use of 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 provide electric power back to the grid, increasing the reliability and stability of the grid. 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 coalition formation possible 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 cheap fossil fuels, like coal, oil and natural gas. However, fossil fuels are non-renewable, highly polluting energy sources whose availability is becoming increasingly scarce. Furthermore, environmental and economic impacts resulting from the use of non-renewable energy sources has received much attention of modern society. In order to reverse this unsustainable scenario, 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 has evolved very little over the years. The resulting scenario is a precarious and inefficient grid, 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 by the U. S. Department of Energy [1] 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. Its distributed intelligence, coupled with broadband communications and automated control systems, enables real-time market transactions and seamless interfaces among people, buildings, industrial plants, generation facilities, and the electric network.

Moreover, this concept is strongly connected to the idea of a more sustainable society based on energy efficiency and on the use of renewable and less aggressive energy sources.

The smart grid concept is closely related to computing, since it was conceived under a solid foundation from engineering, communications, distributed intelligence, automation and information exchange [2]. Furthermore, given its essentially distributed nature, smart grids represent a large field of research for multiagent systems, which are able to enhance the smart grids' operation and generate greater benefits to society.

In addition to the concept of smart grids, recently the concept of vehicle-to-grid (V2G) has been used, in which electric vehicles (EVs) can provide part of the energy stored in their batteries back to the grid when needed [3]. This mechanism becomes important in situations where the grid relies on intermittent renewable energy, as it is the case of energy provided by wind turbines and solar panels. In such cases, the energy stored in the EVs' batteries can be provided back to the grid when production is not able to meet the demand.

The use of V2G mechanisms has received great attention of many researchers who associate agents and smart grids, as in [4], [5], [6], [7], [8] and [9]. In essence, the concept of V2G can be seen as a distributed energy storage system. This scenario has a great potential for research involving social computing. According to Wang et al. [10], social computing is a paradigm interested in facilitating studies of social dynamics through computational mechanisms. One of the theoretical underpinnings of social computing is concerned with agents' organization, which may take place through several paradigms, such as coalition formation [11]. In V2G sessions, given that EVs are unable to operate alone in a cost effective way due to insufficient energy capacity and availability, coalition formation is an efficient way to increase the reliability on EVs' supply process [12].

Coalition formation is a research topic that has received great attention in the field of multiagent systems. According to [13], a coalition can be defined as a group of agents that decide to cooperate in order to achieve a common goal. In many real situations it is possible to save resources through the coordination of activities. 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 [14]. According to [14], coalition formation includes three activities:

1. Coalition structure generation, where the set of agents must be partitioned into disjoint and exhaustive coalitions. This partition is called a coalition structure (CS).
2. Solving the optimization problem of each coalition, where 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 [15], Rahwan et al. proposed a near-optimal coalition structure generation algorithm. Although this algorithm has represented a major breakthrough in the area, it does not yet perform well for a large number of agents. In this and other recent work, the number of agents for which the algorithm performs well is around 20 agents. It must be noted that this approach is generic and can be applied to a larger number of problems. Given a specific application, however, it is possible to use domain information to establish criteria that allows the reduction of the search space for the optimal solution.

In this paper we focus on the reduction of the coalition structure's 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 to reduce the complexity of the process, since using domain information enables modeling the problem to establish criteria to prune the search space for the optimal solution. 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. Furthermore, we use simulations to evaluate different scenarios showing that the more restrictive the constraint and the greater the number of agents, the greater the percentage of the search space to be pruned.

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 and 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

As mentioned earlier, there is a wide variety of research involving multiagent systems and smart grids. Here we review relevant related work of this field, highlighting some key aspects relative to our work. 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 renewable energy generators with small-to-medium energy capacity, like wind turbines, solar panels etc. Taking into account that renewable energy sources are intermittent due to weather conditions, their approach suggests grouping DERs to promote reliability and efficiency on production predictions throughout the day-ahead. Thus, the energy price can be established for a day ahead, ensuring that the market will work properly. The proposed mechanism incentivizes DERs to provide accurate estimates about its production, rewarding the good ones and punishing the bad ones. However, this approach has a greater focus on mechanism design than in coalition structure generation, disregarding how far the solution is from the optimal one.

In the work of Vasirani et al. [5], the focus is on coalition formation among wind turbines and EVs, also forming VPPs. The goal on forming such coalitions is to solve the problem of intermittent power generation of the wind turbines through the use of EVs' batteries, increasing the reliability of this kind of energy and thereby increasing profitability. According to [3], private vehicles are parked 96% of time on average, which greatly enhances the use of EVs in V2G sessions. Therefore, in the proposal of Vasirani et al., wind turbines could improve their profits, whereas EVs could 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.

In Kamboj et al.'s approach [6], coalition formation among EVs is proposed, which can act as suppliers in the regulation market. The goal of the regulation market is to bring stability to the grid by ensuring that it always meets the demand. The regulation market basically provides power to the grid whenever demand exceeds supply and store energy whenever supply exceeds demand. To provide energy, the market usually depends on batteries (can readily store and supply energy, but are very expensive) and generators (can generate energy, but they are very polluting and take some time to start working). Thus, Kamboj et al. propose the use of EVs' batteries for this task. Taking into account that vehicles remain parked most of the time and assuming that they are plugged into the grid during this period, the use of EVs' batteries would help to reduce costs and to improve efficiency of the regulation market. With this approach, the EV may profit by selling energy to the grid and may save money on buying energy when it is cheaper. However, coalition structure generation is made in an ad hoc fashion, without concern for the optimal solution.

More recently [7], Kamboj et al. extended their previous work with new features and results acquired from the deployed version of their system, which is running for five real EVs. In their work, four different agents were modeled: EV, which represents the owner interests; aggregator, which is responsible for forming coalitions; TSO, which represent the grid itself; and charger, which controls a battery charger. Every time an EV is connected to a charger, it begins negotiating with all aggregators in its geographic area, joining the coalition of the aggregator that has made the best offer. Although their work is already deployed in real world, it does not care about finding the optimal coalition structure.

In [9], Mihailescu et al. proposes a dynamic coalition formation-based model that runs efficiently in distributed environments with negotiation agents. In this proposal, coalitions are made of DERs, which can assume two possible roles: provider, when the DER can sell its surplus energy; and consumer, when it needs to buy energy. In this approach, coalition formation consists of three phases: coalition initiation, where some agents, whose energy availability exceeds a pre-defined value, are probabilistically selected to coordinate the coalitions; provider aggregation, where coordinator agents negotiate with provider agents to form coalitions; and the consumer aggregation, where consumer agents are included in coalitions based on its proximity. This approach, however, does not guarantee that the formed coalition structure is the optimal one.

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. As mentioned previously, a coalition can be defined as a group of agents that decide to cooperate in order to achieve a common goal [13]. A coalition structure is a partition of the set of agents into disjoint and exhaustive coalitions [14]. As is common practice in the literature [13, 15, 14], coalition formation is studied as characteristic function games (CFG), where the value of each coalition C is given by $v(C)$ [16]. The value v represents how beneficial the formation of such a coalition could be. In CFGs, the characteristic function has an important role because it enables modeling the problem with respect to the constraints of the domain.

Since the focus of this research topic is not on generating any coalition structure but on finding the best one, it becomes clear how big the problem is: given a set of a agents, the number of possible coalitions is $2^a - 1$ and of coalition structures is $O(a^a)$ and $\omega(a^{\frac{a}{2}})$ [14]. The search space of coalition structures can be represented by a graph, as in Fig. 1.

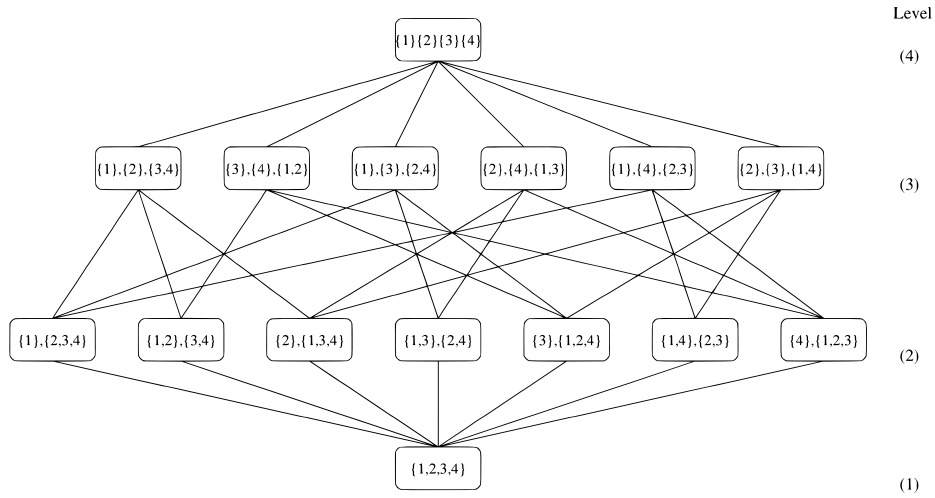


Fig. 1. Coalition structure graph for four agents (extracted from [14])

As shown in Fig. 1, coalition structures are grouped in levels according to the number of coalition they have, e.g., coalition structures that have two coalitions

are located in the second level of the graph. The number of coalition structures of the graph level l can be obtained by the Stirling number of the second kind, which can be calculated through the following recurrence [14]:

$$Z(a, l) = lZ(a - 1, l) + Z(a - 1, l - 1) , \quad (1)$$

where $Z(a, a) = Z(a, 1) = 1$. Additionally, the exact number of coalition structures can be obtained by summing up the resulting Stirling number of all graph levels, as in

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

Finding the optimal coalition structure can be seen as searching in the coalition structure graph, which is unfeasible due to computational complexity. In [14], Sandholm et al. 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 [15], Rahwan et al. proposed an anytime algorithm to find optimal coalition structures. To understand how it works, some basic definitions are necessary. The set of all 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 $V(CS) = \sum_{C \in CS} v(C)$ represents the sum of the values of each coalition that comprise it. The algorithm is interested in finding the optimal coalition structure $CS^* = \arg \max_{CS \in \mathcal{CS}} V(CS)$, that is, the one with the greatest value.

The proposed algorithm in [15] is based on a novel representation of the search space, where coalition structures are grouped by the size of coalitions they have (called configuration). 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 $s \in \{1, 2, \dots, a\}$. And let $G_1, G_2, \dots, G_{|\mathcal{G}_{CS}|} \in \mathcal{G}_{CS}$ be the set of all possible configurations. Thus, $F(\mathcal{G}_{CS})$ is a function that returns all the coalition structures that follow that configuration. Finally, $\mathcal{N} = \{F(G_1), F(G_2), \dots, F(G_{|\mathcal{G}_{CS}|})\}$.

Based on those definitions, the algorithm in [15] 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. Although this stage involves searching all possible coalitions, the search space for the coalition structures is much greater.
2. Choosing the optimal configuration: uses the values obtained in the previous stage to generate the list $G \in \mathcal{G}_{CS}$ and find the element $F(G)$ to search for the optimal solution CS^* . This stage identifies only one optimal configuration, which allows performing a search in only a small portion of the coalition structure's search space in the next stage.

3. Finding the CS^* : where all $V(CS) \in F(G)$ are computed to find the optimal solution CS^* . This is the most computationally costly stage.

After performing these three stages, the algorithm returns the optimal solution having searched only $3 \times 2^{a-1}$ coalition structures, an extremely small portion of the search space.

4 Problem Modeling

The scenario presented in our work is a smart grid where EVs sell their surplus energy on V2G sessions when supply does not meet demand. In this scenario, the grid incentivizes these EVs to form coalitions by offering more attractive prices when energy supply is greater. As common in the literature [4, 9, 12], coalitions should form small scale VPPs, i.e., VPPs whose energy capacity is around 1 MW. Coalitions are formed at the beginning of a time step $t \in \mathcal{T}$ and lasts until the end of that period. As in [4], each time step t corresponds to a half-hour period. Formally, this system is assumed to be a closed world so that no agent would enter or leave it early, i.e., coalition structures remain intact from the beginning to the end of each time step.

The formalism of this problem is addressed in the form of CFGs, as in [13, 15, 14]. 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 the coalition $v(C)$ would be equals to 0.05. In the same way, if the coalition had only one agent, the grid would pay only \$1.00 per kWh, that is, $v(C)$ in this case would be zero.

In this paper we have defined only one kind of agent, the EV. The aim of this agent is to sell the surplus energy on its battery to the grid, getting the highest profit as possible. The internal battery of each EV has an amount of surplus energy $w_{\text{available}}$ that can be sold on V2G sessions.

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: EVs should supply energy only to consumers who are in the same region they are. Otherwise, if the supplied energy travels long distances to 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.

With the purpose of accomplish the distance constraint, the grid can define a maximum distance α among agents of the same coalition. Thus, let $D \subseteq A$ be a set of agents that cannot form coalitions among themselves due to the distance constraint. Here it is important to take into account that 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 be in the same coalition and also prevent agents $a3$ and $a4$ to do so, then two different sets D would exist: $D_1 = \{a1, a2\}$ and $D_2 = \{a3, a4\}$; based on this constraints, two possible coalitions could be formed: $\{a1, a3\}$ and $\{a2, a4\}$. Now, suppose there is one more agent $a5$, which cannot form coalitions with agents $a1$ and $a3$; thus, a new set $D_3 = \{a1, a3, a5\}$ must be created. Hence, let $\mathbb{D} = \{D_1, D_2, \dots, D_{|\mathbb{D}|}\}$ be the set of all possible D sets.

Our technique works by selecting the greatest set $D^* = \arg \max_{D \in \mathbb{D}} |D|$ 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 $D^* = \{a1, a2, a3\}$, i.e., two levels can be pruned. Then, in order to keep the D^* 's agents separated, a coalition structure must have at least three coalitions; thus, the two 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, which is divided into two stages. First, for each agent $a \in A$ its own set of constraints is identified, i.e., the set with all agents that cannot form coalitions with a . These sets are then analyzed in the second stage to find the D^* . For this purpose, the algorithm requires domain information, namely, the distance among the agents. Thus, let d_b^a be the distance between the agents a and $b \in A$. Additionally, as described previously, let α be the distance constraint existing in the grid.

In the first stage of the algorithm, for each agent $a \in A$ a $D^a \in \mathcal{D}$ set is created, which is populated with every agent b whose distance $d_b^a > \alpha$, as can be seen at line 5 of Alg. 1. Note that \mathcal{D} and \mathbb{D} are essentially different: all agents in $D^a \in \mathcal{D}$ have constraints with a , but not necessarily among themselves. On the other hand, all agents of a given $D \in \mathbb{D}$ have constraints among themselves. Thus, based on \mathcal{D} , the algorithm is able to find the $D^* \in \mathbb{D}$. Also in the first stage, the algorithm calculates $g = \max_{D^a \in \mathcal{D}} |D^a|$ for optimization purposes in the second stage.

In the second stage, all D^a sets are analyzed to identify constraints among the agents inside of them. In other words, for each D^a set, the algorithm must identify the agent that has the greatest number of constraints with the other agents of the same set. In order to accomplish this task, 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 . This is done by the recursive function at lines 25 to 36 of Alg. 1. The result calculated by the recursive function is assigned to s_D (line 13 of Alg. 1). To avoid the search on redundant D^a sets while finding the D^* , the algorithm considers only agents greater than b when a D^b set is being created, as can be seen at line 28. This process ends only after all $D^a \in \mathcal{D}$ were analyzed or when a D^b set whose size is greater than g

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

Requires: A : the set of agents
 : d_b^a for all $a \in A$ and $b \in A/a$: the distance among each agents
 : α : the distance constraint
Returns : the size of D^*

```

1 function findBestD ()
2 begin
3   initialize  $s \leftarrow 0, g \leftarrow 0$ ;
4   /* computes the set of constraints  $D^a$  of each agent  $a$  */
5   foreach  $a \in A$  do
6     | let  $D^a \in \mathcal{D} \leftarrow \arg_{b \in A/a} d_b^a > \alpha$ ;
7   endforeach
8   let  $g \leftarrow \max_{D^a \in \mathcal{D}} |D^a|$ 
9   /* iterates over all  $D^a$  to identify the size of  $D^*$  */
10  foreach  $D^a \in \mathcal{D}$  do
11    | foreach  $b \in D^a$  do
12      | if  $|D^a| == 1$  then
13        |    $s_D \leftarrow 2$ ;
14      | else
15        |    $s_D \leftarrow \text{findDs}(D^a) + 1$ ;
16        |    $D^a \leftarrow D^a / b$ ;
17      | endif
18      | if  $s_D > g$  then
19        |   return  $s_D$ ;
20      | else if  $s_D > s$  then
21        |    $s \leftarrow s_D$ ;
22      | endif
23    | endforeach
24  endforeach
25  return  $s$ ;
26 end findBestD;

/* recursively iterates over a given  $D^a$  to identify the agents that
   have constraints among themselves */
27 function findDs( $D^a$ )
28 begin
29   let  $b \leftarrow$  first element of  $D^a$ ;
30   let  $D^b \leftarrow \arg_{c \in D^a / b, c > b} d_c^b > \alpha$ ;
31   if  $|D^b| = 1$  then
32     | return 2;
33   else if  $|D^b| > 1$  then
34     | return  $\text{findDs}(D^b) + 1$ ;
35   else
36     | return 1;
37   endif
38 end findDs;
```

($s_D > g$) has been encountered (line 16 of Alg. 1). The stopping criteria defined by g interrupts the search if all agents of the greatest D^a have constraints among themselves, meaning that this is the D^* .

The following examples help to understand how this technique works. Example 1 presents a simple scenario for four agents. Example 2 presents a more specific scenario in which the stop criteria defined by g is achieved.

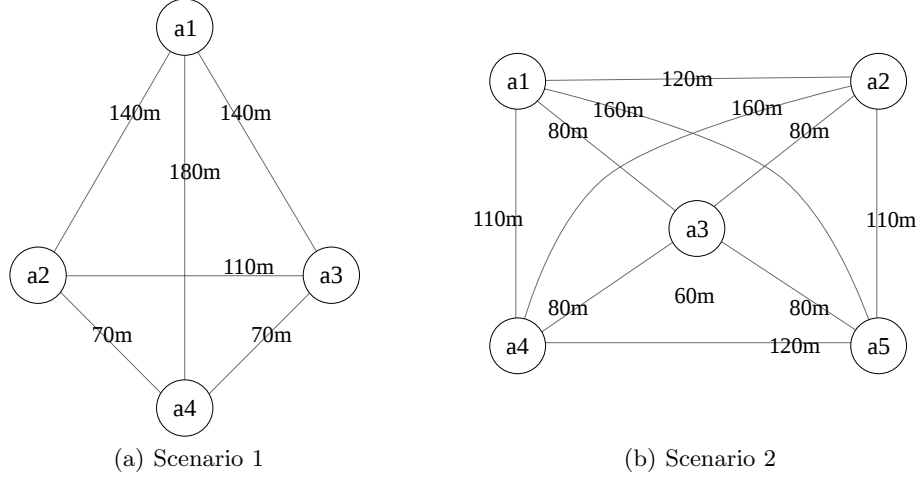


Fig. 2. Hypothetical scenarios of examples 1 and 2

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

In this example, agent $a1$ cannot be in the same coalition as agents $a2$, $a3$ and $a4$ because its distance to them is greater than α , meaning that $D^{a1} = \{a2, a3, a4\}$. Similarly, for agents $a2$ and $a3$ we have $D^{a2} = \{a1, a3\}$ and $D^{a3} = \{a1, a2\}$. Finally, agent $a4$ have constraints with agent $a1$ only, that is, $D^{a4} = \{a1\}$. Therefore, after the first stage of the algorithm we have $\mathcal{D} = \{\{a2, a3, a4\}, \{a1, a3\}, \{a1, a2\}, \{a1\}\}$ and $g = 3$. In the second stage, the algorithm iterates over every $D^a \in \mathcal{D}$. The result is $D^* = \{a1, a2, a3\}$, meaning that the two 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 three partitions on coalition structures are required. This happens only in the two highest levels. Based on Eq. 2, there are 15 possible coalition structures for four agents. Similarly, the size of the two lowest levels of the graph can be calculated with Eq. 1, resulting in 1 and 7 coalition structures, respectively. Therefore, this pruning corresponds to 53.33% of the search space.

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

In the scenario of this example, the agents $a1$, $a2$, $a4$ and $a5$ can not form coalitions among themselves due to the distance constraint. Therefore, $D^{a1} = \{a2, a4, a5\}$, $D^{a2} = \{a1, a4, a5\}$, $D^{a4} = \{a1, a2, a5\}$ and $D^{a5} = \{a1, a2, a4\}$. On the other hand, agent $a3$ has no constraints with the other agents, resulting in $D^{a3} = \emptyset$. Therefore, $\mathcal{D} = \{\{a2, a4, a5\}, \{a1, a4, a5\}, \emptyset, \{a1, a2, a5\}, \{a1, a2, a4\}\}$ and $g = 3$. In the first iteration of the second stage, D^{a1} is analyzed. Since all agents inside of D^{a1} have constraints among themselves, the set $D_1 = \{a1, a2, a4, a5\}$ has been found. Considering that $|D_1| > g$, the stopping criteria of line 18 of Alg. 1 is achieved. Here, it is clear that since the D_1 is the greatest possible D , no further search is required. Therefore, we have $D^* = \{a1, a2, a4, a5\}$, meaning that three levels can be pruned. The search space in this scenario have 52 coalition structures. By pruning three levels, 41 coalition structures are removed, representing a prune of approximately 78.84% 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 among the agents, the D^* set would be very small. This would reduce the effectiveness of the technique.

From the implementation point of view, this technique can be included on Rahwan et al. algorithm [15] after the pre-processing stage, pruning all configurations that corresponds to the levels to be pruned. With this change, the algorithm could search only configurations of non-pruned levels.

6 Evaluation

In this section we present the evaluation of our technique. We begin by comparing it to the one proposed by Rahwan et al. [15]. In their work, results were presented showing the percentage of space that was searched by their algorithm for 20 agents at most. We now use these results to compare their algorithm with our technique. For this purpose we generated results for 10, 15 and 20 agents, varying the size of the $|D^*|$ set. By doing so, we can compare the percentage pruned by our technique with theirs.

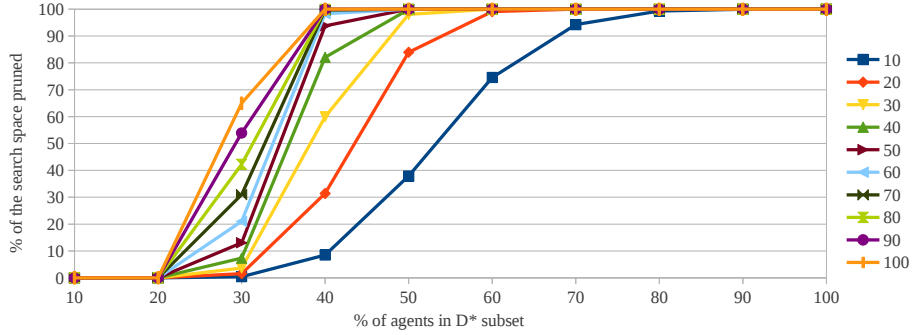
As shown in Table 1, the algorithm proposed by Rahwan et al [15] performs increasingly better as the number of agents increases. However, it is not able to run for more than 20 agents. Similarly, our technique performs better as the number of agents increases. Nevertheless, it performs better than the Rahwan et al. one for a small number of agents and it is able to run for more than 20 agents.

Since our intention is to use domain information in order to prune the search space and to increase the number of agents, we now show results for a greater number of agents. Figure 3 presents a graph that shows how much of the search space can be pruned as the number of agents in the D^* set increase. Results were generated for different number of agents from 10 to 100 (only multiples of

Table 1. Pruned percentage of the search space by our mechanism compared to the one of Rahwan et al. [15]

a	$ CS $	$ D^* $	Pruned CSs	% pruned	% pruned by [15]
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

ten). For each number of agents, we present how much space may be pruned according to the percentage of agents in the D^* set (also multiples of ten, only).

**Fig. 3.** Pruned percentage of the search space for 10 to 100 agents

As can be seen 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% sooner as the number of agents increase.

In Fig. 4 the same results are presented for 100 to 1000 agents (for multiples of hundred). As can be seen in the graphs, 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 space is small. In the case of a thousand agents, the entire

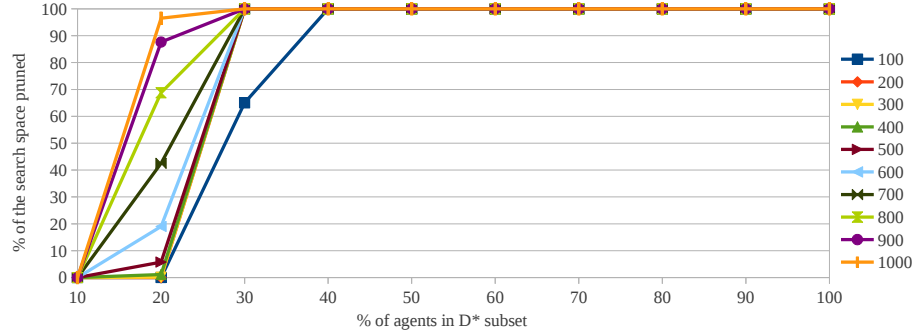


Fig. 4. Pruned percentage of the search space for 100 to 1000 agents

search space has the order of 10^{1927} coalition structures and, for $|D^*| = 30\%$, the remaining search space has the order of 10^{1857} .

The previous discussion has shown that the coalition structure's search space can be drastically pruned as the size of D^* increases. We now 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 on V2G sessions. This scenario is illustrated in Fig. 5 and the distance among the agents is in Fig. 6.

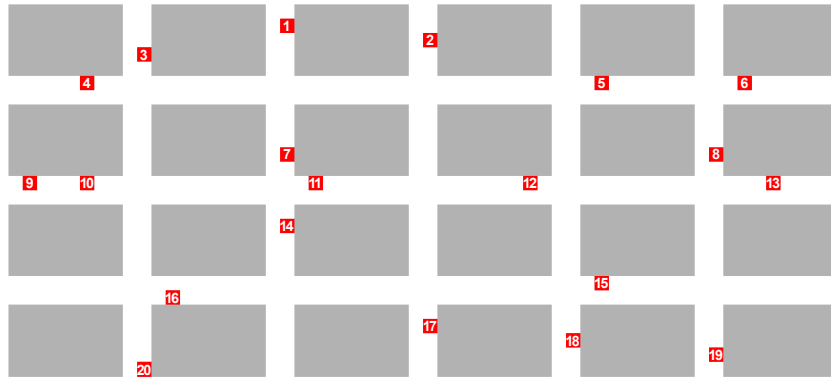


Fig. 5. Hypothetical scenario of a neighborhood with 20 EVs plugged into the grid

In this hypothetical scenario, each EV has its own energy availability $w_{\text{available}}$, which is selected with uniform probability from 0.5 to 2 kWh. The distance constraint of the grid is $\alpha = 100\text{m}$. Each unit of energy, whose regular price is \$0.50, corresponds to 1 kWh. The coalition values $v(C)$ were generated based on the size of the coalitions (i.e., $|C|$) and on the amount of energy available on

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. 6. Distance (in meters) among the agents of the hypothetical scenario of Fig. 5

it. Based on these informations, the algorithm identified the greatest constraints set as $|D^*| = 11$, meaning that 10 levels of the graph may be pruned.

In this scenario, for 20 agents there are $\approx 5.17 \times 10^{13}$ coalition structures. After pruning 10 levels of the graph, only $\approx 4.6\%$ has remained. Based on that, the algorithm was able to find the optimal coalition structure $CS^* = \{\{1\}, \{2\}, \{17\}, \{20\}, \{3, 4\}, \{7, 11\}, \{8, 13\}, \{9, 10\}, \{14, 16\}, \{5, 6, 12\}, \{15, 18, 19\}\}$, which has the configuration $\{1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3\}$, and whose value is $V(CS^*) = 0.216$.

For the sake of comparison, the Rahwan et al. algorithm [15] would need to search on $\approx 50\%$ of all coalition structures to find the optimal one. On the other hand, after being extended with our technique, the algorithm needs to perform the search on $\approx 4.6\%$ of the search space only. Naturally, the performance of our technique, in terms of pruning, is strongly related to the value of α , i.e., the higher the value of α (a more restrictive α) the smaller the search space, as depicted in Fig. 7. However, since our algorithm is an extension of Rahwan et al. algorithm, in the worst case (a less restrictive α), the performance of our algorithm would be equal to the Rahwan et al. algorithm, whereas in the best case it would prune twice as much.

7 Conclusions

In this paper we presented an algorithm for finding the optimal coalition structure. The algorithm focuses on identifying agents that cannot form coalitions among themselves, based on domain information, to perform a considerable prune on the coalition structure's search space. This technique may be applied in smart grids to improve V2G mechanisms. In this scenario, due to specific

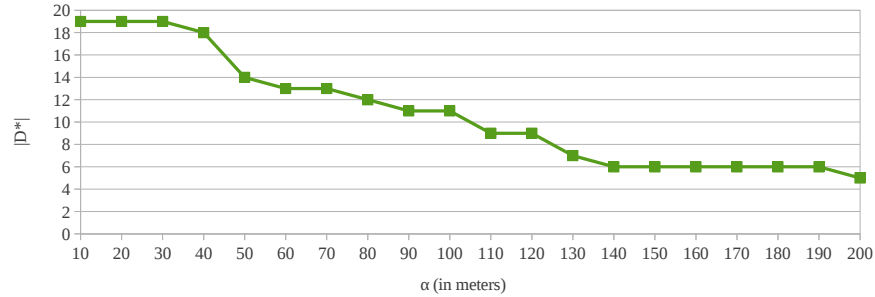


Fig. 7. $|D^*|$ for different values of α in the scenario of Fig. 5

constraints of the grid, our technique has shown to be effective, performing a good pruning on the coalition structure's search space.

For future work we expect to improve our technique to prune not only entire levels, but also specific portions of a such level. To achieve this we may build on the representation proposed in [15], 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 one by Bazzan and Dahmen [13], where the idea of more valuable players was proposed, that is, agents which aggregate additional value to the coalitions they join. Based on this concept, it is possible to identify the level where the optimal coalition is, without the need to perform a search in the whole search space. Here we expect to identify elements of smart grids which could aggregate additional value, like buildings whose energy capacity is supposed to be greater than the EVs. From smart grid's point of view, buildings may represent an attractive energy source given that they usually may have a lot of EVs parked in, and even their own generators, like wind turbines or solar panels. Finally, it would be interesting to explore ways of distributing the search among the agents in order to parallelize the process and make it faster.

Acknowledgements

We are very grateful to the anonymous reviewers for their valuable suggestions. We also thank Anderson Tavares and Cássio Oliveira for their helpful comments. The authors are partially supported by Brazilian National Counsel of Technological and Scientific Development (CNPq) and Foundation for Research Support of the State of Rio Grande do Sul (FAPERGS).

References

1. U. S. Department of Energy: Grid 2030: A national vision for electricity's second 100 years (July 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 (May 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: *The Tenth International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2011)*. (May 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 (ATES 2010)*. (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 (May 2011) 13–20
8. Matta, N., Rahim-Amoud, R., Merghem-Boulahia, L., Jrad, A.: A cooperative aggregation-based architecture for vehicle-to-grid communications. In: *Global Information Infrastructure Symposium (GIIS)*, 2011, IEEE (2011) 1–6
9. Mihailescu, R.C., Vasirani, M., Ossowski, S.: An organizational approach to agent-based virtual power stations via coalitional games. In Pérez, J., Corchado, J., Moreno, M., Julián, V., Mathieu, P., Canada-Bago, J., Ortega, A., Caballero, A., eds.: *Highlights in Practical Applications of Agents and Multiagent Systems*. Volume 89 of *Advances in Intelligent and Soft Computing*. Springer Berlin / Heidelberg (2011) 125–134
10. Wang, F.Y., Carley, K.M., Zeng, D., Mao, W.: Social computing: From social informatics to social intelligence. *IEEE Intelligent Systems* **22**(2) (2007) 79–83
11. Horling, B., Lesser, V.: A survey of multi-agent organizational paradigms. *Knowledge Engineering Review* **19**(4) (2004) 281–316
12. Pudjianto, D., Ramsay, C., Strbac, G.: Virtual power plant and system integration of distributed energy resources. *Renewable Power Generation, IET* **1**(1) (March 2007) 10–16
13. 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 (December 2007) 180–192
14. 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
15. 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
16. Schmeidler, D.: The nucleolus of a characteristic function game. *SIAM Journal on applied mathematics* **17**(6) (1969) 1163–1170