A Self-Adapting Similarity-Based Coalition Formation Approach for Plug-in Electric Vehicles in Smart Grids

Gabriel de O. Ramos
Instituto de Informática
Universidade Federal do Rio Grande do Sul
goramos@inf.ufrgs.br

Juan C. Burguillo
Telematics Engineering Department
University of Vigo
j.c.burguillo@uvigo.es

Ana L. C. Bazzan
Instituto de Informática
Universidade Federal do Rio Grande do Sul
bazzan@inf.ufrgs.br

Abstract

The Vehicle-To-Grid (V2G) concept is a key feature towards the integration of electric vehicles into smart grids. Through V2G sessions, plug-in electric vehicles (PEVs) can sell their surplus energy to the grid. However, profiting from V2G sessions is not trivial for singletons. Thereby, the formation of coalitions among PEVs has been proposed to tackle this issue. In this paper, we rely on domain knowledge in order to propose a novel modelling for such problem. Specifically, we investigate how the similarity among the PEVs’ energy profiles can be used to improve the formation of coalitions. The energy profile of a PEV estimates how long such PEV will be available for the V2G session. Based on this, we aim at maximising the coalitions’ duration. We empirically show that our approach is both efficient (it outperforms state-of-the-art algorithms in terms of runtime) and effective (solutions were 96.5% of optimum, on average).

Keywords: Game theory, coalition formation, smart grid, vehicle-to-grid

1 Introduction

Essential services, including telecommunications, transportation, and industrial activities, strongly depend on electricity. Such a fundamental resource is provided by means
of electricity networks (grids). The grid is the infrastructure responsible for connecting energy producers and consumers. Conversely, grids have evolved very little in spite of their importance. The concept of smart grid emerges in this scenario: a fully automated electricity network that intensively monitors and controls its elements, being able to supply energy in an efficient, reliable way [21]. A key feature of a smart grid is the bidirectional flow of energy and communication between its elements, i.e., every element can both supply and consume energy.

Many efforts towards smart grids have focused on distributed energy resources. An interesting concept here is the so-called Vehicle-To-Grid (V2G), through which plug-in electric vehicles (PEVs) can provide part of the energy available in their batteries to the grid [7, 6]. The V2G mechanism is particularly useful when the grid relies on intermittent, renewable energy sources, such as wind and solar.

The participation of PEVs on V2G sessions in a cost-effective way is far from trivial. According to Pudjianto et al. [9], when acting in V2G sessions, PEVs must commit to provide a given amount of energy. However, single PEVs may not manage to fulfil their commitment to the grid. Therefore, many studies have proposed grouping PEVs in the so-called virtual power plants (VPPs), through which PEVs can provide energy in a more predictable, reliable way. Coalition formation has shown to be a suitable alternative to this end [15].

A coalition can be defined as a group of agents that decide to cooperate in order to achieve a common goal. According to Sandholm et al. [19], coalition formation includes three basic activities: coalition structure generation (CSG), solving the optimization problem of each coalition, and dividing of the obtained payoff among the agents. In this work, we focus on the former task.

The CSG activity is an optimization problem whose aim is to find the partition that yields the highest utility. Such activity has been proven to be NP-complete [19], and traditional CSG algorithms [13, 10, 20, 19] do not scale and do not deal with dynamically changing scenarios (i.e., agents can enter or leave the system). In this sense, our focus is not on finding the optimal solution, but on finding a good one, in a reasonable amount of time, while also considering dynamic scenarios.

In previous works [17, 18], we have investigated how constraints present in the domain of smart grids might be used to improve the coalition formation process. Although good results have been achieved, other kinds of relations among the PEVs can be explored [9, 8, 4]. Thus, in this paper we propose an extension by investigating how the similarity among PEVs’ energy profiles can be used to improve the formation of coalitions. The energy profile of a PEV represents its availability for V2G sessions. Through such a modelling, we aim at promoting lasting, stable coalitions.

We present a distributed heuristic-based method for CSG in dynamic environments, called Self-Adapting Coalition Formation (SACF henceforth). The problem is modelled as graph, where: (i) agents are represented by nodes, (ii) links exist between agents whose energy profile is similar, and (iii) a feasible coalition must be a complete subgraph. In SACF, neighbouring agents negotiate the formation of feasible coalitions among themselves. As a result, infeasible coalition structures are pruned from the search space, thus allowing SACF to mitigate scalability issues faced by classical CSG approaches. Moreover, as opposed to our work presented in [17], here PEVs are allowed to check for better coalitions and negotiate their formation. Such a self-adapting
capability enables agents to improve their revenue whenever possible.

Through experiments, we show that SACF outperformed state-of-the-art CSG algorithms in terms of runtime by many orders of magnitude. The average solution quality neared 96.5% of the optimal (with a standard deviation of 5.5%) in all tested cases. In dynamic environments, the agents’ payoff has shown reasonably stable over time. Moreover, the payoff was higher for in-coalition than for single PEVs. However, our approach has the drawback of not providing guarantees on the quality of solutions.

This paper is organised as follows. The next section introduces some basic definitions on coalition formation. An overview of the literature is presented in Section 3. The problem modelling and our method are presented in sections 4 and 5. Our approach is empirically evaluated in Section 6. Finally, conclusions and future directions are presented in Section 7.

2 Preliminaries

A coalition is a group of agents that decide to cooperate in order to achieve a common goal, aiming at improving their performance [20]. Given a set of agents \( \mathcal{A} = \{1, 2, \ldots, n\} \), a coalition is a subset \( C \subseteq \mathcal{A} \). A coalition structure is a partition \( CS = \{C_1, C_2, \ldots, C_{|CS|}\} \) of the set of agents into disjoint (i.e., \( \forall C_i, C_j \in CS, C_i \neq C_j, C_i \cap C_j = \emptyset \)) and exhaustive (i.e., \( \bigcup_{C \in CS} C = \mathcal{A} \)) coalitions.

Coalition formation is usually studied in the form of characteristic function games (CFGs). In CFGs, a characteristic function \( v : 2^n \rightarrow \mathbb{R} \) assigns a value \( v(C) \) to each possible coalition \( C \subseteq \mathcal{A} \). The value of a coalition structure \( CS \) can be obtained by summing up the values of the coalitions that compose it, i.e., \( V(CS) = \sum_{C \in CS} v(C) \).

As previously discussed, the coalition formation process comprises three activities. In this paper, we focus on the coalition structure generation (CSG) activity, which regards the formation of coalitions per se. In CSG, one aims at finding the optimal coalition structure \( CS^* = \arg \max_{CS \in CS} V(CS) \), i.e., the one with the highest value. However, there is a scalability issue: the number of possible coalitions is \( 2^n - 1 \) and of coalition structures is asymptotically in the order of \( O(n^n) \) and \( \omega(n^{2n}) \). Furthermore, this problem has proven to be NP-complete [19].

3 Literature Review

A number of approaches have been proposed to solve the CSG problem. One of the most outstanding methods was proposed by Rahwan et al. [13], which is based on integer-partition, and thus known as IP. In their approach, an efficient solution space representation was proposed, where coalitions are grouped by their sizes and coalition structures are grouped by the size of coalitions they have. Based on such representation, the IP algorithm makes use of the branch-and-bound paradigm to provide anytime approximate solutions. Although relatively efficient, the time complexity is high: in the worst case, \( O(n^n) \) coalition structures have to be searched. As a result, IP is not scalable for more than a few agents (e.g., finding the optimal solution for 20 agents takes many hours).
Another important approach was proposed by Rahwan and Jennings [10], which makes use of the dynamic programming paradigm. The Improved Dynamic Programming (IDP) algorithm is based on the dynamic programming algorithm of Yeh [26] for the set partitioning problem. Basically, IDP works by deciding for each coalition whether it is better to split it into two small coalitions or to keep it as it is. To this end, the algorithm needs to compute and store the solution and value of all possible coalitions. The IDP’s worst case computational complexity is $O(3^n)$, which is much better than IP. However, IDP has a memory requirement of $O(2^n)$ and is not anytime.

Clearly, despite being optimal and generic, both IP and IDP are too slow to be considered for realistic applications. In this sense, many studies have considered exploring the problem’s structure.

For instance, in a previous work, we have proposed an improved pre-processing phase for IP, where domain information is used in order to prune infeasible coalitions from the search space [16]. In the worst case, however, the time complexity remains $O(n^n)$. In [11], a framework for a particular class of constrained coalition formation (CCF) problems is proposed. In CCF, constraints are imposed on the CSG process through propositional logic statements, and the problem is solved through a divide-and-conquer algorithm. Their approach, however, cannot model the kind of constraints used here. Compact representations of the characteristic function were studied by Ueda et al. [22] and Rahwan et al. [12]. Such approaches exploit the problem’s structure in order to make the problem representation more compact. However, they usually provide lower quality solutions.

Coalition formation in synergy graphs was studied by Voice et al. [25]. In a synergy graph, nodes represent agents, edges represent a kind of relation among them (e.g., trust), and a coalition to be feasible must be a connected graph. In this sense, an IDP-based algorithm was proposed to solve the CSG problem. Although faster than IDP (under certain conditions), their approach does not scale well and is not fast enough for real situations. An improvement was proposed by Bistaffa et al. [1], in which the concept of edge contraction was included. Although their approach scales for a higher number of agents, it remains slow for coping with dynamic, real scenarios.

Other approaches include the work of Farinelli et al. [4], which addresses the CSG problem through a hierarchical clustering algorithm. However, although it may run for many agents, the solution’s quality is not as good as other approaches. The formation of coalitions in task-oriented domains was studied in the work of Ramchurn et al. [14]. Their approach, nonetheless, is more concerned with solving the tasks (not finding the best coalitions), and is suitable for up to 20 agents. Chalkiadakis and Boutilier [2] have proposed a Bayesian model-based reinforcement learning framework for repeated coalition formation under uncertainty. Such approach, however, is more concerned with agents’ learning and decision making, and does not address the CSG problem.

The use of coalitions in smart grids has been widely investigated [15]. One of the main interests of the field has been to increase the reliability of renewable energy production. Chalkiadakis et al. [3] addressed the formation of VPPs of wind turbines. In [5], Kamboj et al. investigate the formation of coalitions among PEVs to stabilise the grid’s energy supply. The formation of coalitions among energy producers and consumers was addressed by Mihailescu et al. [8] and Vasirani et al. [23]. In the former, producers and consumers with similar energy profiles are grouped. On the
other hand, in [23], PEVs are used to mitigate the intermittence of the wind turbines supply. However, all these works handle coalition formation in an ad hoc fashion, disregarding the quality of the coalitions.

As can be seen, there is still a lack of approaches to handle coalition formation in dynamic, realistic scenarios. On the one hand, there are algorithms that generate optimal solutions but are too slow for practical scenarios. On the other hand, there are fast algorithms, whose solutions are not good enough. Moreover, most works regarding VPPs do not address coalition formation properly.

In this context, we investigated how constraints present in the smart grids domain might be used to improve the formation of coalitions of PEVs [17, 18]. Specifically, we considered the geographical distance among PEVs as a determinant factor in the feasibility of coalitions. The difference between [17] and [18] lies in the negotiation scheme, where single PEVs ask or are invited to join existing coalitions, respectively. By modelling such a constraint, we aimed at preventing energy flows to overlap so as to avoid overloading the grid. Although good results have been achieved, such an overlapping is usually prevented by the grid operator in the load flow calculation [9]. Moreover, literature has shown that exploiting other kinds of relations among PEVs is more realistic [4, 8, 9, 24]. Therefore, without loss of generality, in this work the problem is addressed from a different perspective. More precisely, we consider the similarity of energy profiles rather than the geographical distance among PEVs. The aim is to deliver a more reliable energy source by means of lasting coalitions. Our work also diverges from [17, 18] by allowing agents to move to another coalitions whenever a better profit is attainable.

4 Problem Modelling

In this section, we describe the smart grid problem and introduce a novel modelling for it. The scenario considered in this work consists in a smart grid where PEV-agents sell their surplus energy in V2G sessions. Our approach basically works as follows. Whenever a PEV begins a V2G session, a neighbourhood relation is created with all other PEVs whose energy profile is similar. At this point, the PEV starts negotiating to join the best coalition available on its neighbourhood. Once formed, a coalition lasts until one of its members decides to leave. Agents, in turn, can leave their coalitions when a better one has been found or when their time availability has finished (time to unplug).

4.1 Smart Grid Scenario

In this work, the grid incentivises the formation of coalitions among PEVs through a monetary value, which is proportional to the coalition’s power rating and duration, up to certain limits. We assume that the grid is always willing to buy the energy offered by PEVs, i.e., whenever a PEV has energy to sell, the grid will buy it. Furthermore, we only consider PEVs that are willing to participate in V2G sessions.

Although this approach focus on a particular constraint, namely the similarity among energy profiles, it is abstract and powerful enough to accommodate one or more constraints, as evidenced in [17, 18].
Regarding the dynamic aspect of the problem, PEVs can enter or leave the system at any time. This aspect is not just a modelling definition, but a real feature of the domain. The permanency of PEVs in the grid is ruled by their owners’ preferences. To this regard, PEVs should ensure a minimum energy reserve in order to meet their owners’ demand. Thus, we assume that PEV-agents know when to stop selling energy to the grid, i.e., PEVs are allowed to leave the system as soon as they deem necessary.

4.2 Modelling Constraints

As previously discussed in Section 3, traditional CSG approaches have typically addressed abstract scenarios where every set of agents is considered a feasible coalition. However, many (if not most) real-world problems cannot be modelled without taking its constraints into account. Such constraints may be due to communication or physical infrastructures, social or trust relationships, among others. In fact, many recent works have studied how such constraints might be handled in the CSG process [25, 1, 16, 18].

In this work, we assume that a feasible coalition must have PEVs with similar energy profiles. The energy profile of a PEV roughly refers to an estimation of how long (in terms of its plug-in and unplug times) it will be available for the V2G session. The rationale behind such a modelling is that the grid incentivises the coalitions to last as long as possible. However, recall that coalitions last until any of their members decide to leave it. In this sense, in order to maximise the duration of their coalitions, PEVs are better off forming coalitions with peers whose energy profile is similar to theirs, i.e., those peers that are not willing to leave the grid before them. Grouping agents by their energy profile aims at increasing coalitions’ duration, and thus the PEVs profit. From the grid viewpoint, such a modelling translates into a more reliable, stable (coalition negotiation is less frequent) energy source.

Based on this, we model the energy profile of an agent as the amount of time it will be available for the V2G session. The energy profile of a given agent \( i \in A \) is represented by \( T_i \). Recall that agents within a coalition should have similar energy profiles. In this sense, given two agents \( i, j \in A \) with \( i \neq j \), the similarity \( s(T_i, T_j) \) between their energy profiles \( T_i \) and \( T_j \) is measured by the absolute difference between them, as in Equation (1), where the lower the result, the higher the similarity of the given energy profiles. Given such a measure, we can define the neighbourhood relation among the agents. To this end, we constrain the maximum allowed difference between two energy profiles by means of parameter \( \alpha \). Specifically, given two agents \( i, j \in A \) with \( i \neq j \), they are considered neighbours if \( s(T_i, T_j) \leq \alpha \), as formulated by Definition 1.

\[
s(T_i, T_j) = |T_i - T_j|
\]

Definition 1 (neighbourhood) The neighbourhood of a given agent \( i \in A \) is composed of all agents \( j \in A \setminus \{i\} \) for whom \( s(T_i, T_j) \leq \alpha \).

Following this line, a coalition is said feasible if all of its members are neighbours of each other, as in Definition 2. The value of parameter \( \alpha \) should be defined in order

---

2 In this work, each time step corresponds to 1 minute.
to represent the grid’s requirements. A lower value for $\alpha$ is more restrictive than a higher one, meaning that the coalitions are supposed to last longer, at the cost of being smaller.

**Definition 2 (feasible coalition)** A feasible coalition is composed of neighbouring agents, that is, $\forall i, j \in C, i \neq j \ (s(T_i, T_j) \leq \alpha)$.

### 4.3 Communication Layer

Different from traditional CSG approaches, in this work, coalitions are formed by means of local interactions among the agents. In other words, there is not a central authority responsible for regulating the coalition formation process. In this sense, a communication channel among the agents is required. Here we simplify such a requirement through the use of peer-to-peer (P2P) networks. Basically, a P2P network consists of a hub and many leaves connected to it (consider the Gnutella2 protocol, for instance). Such a topology can be seen as analogous to the smart grid scenario considered here: leaves represent PEVs, and hubs are managed by electrical substations (that control the portion of the distribution network where the PEVs are in). The hub is responsible for providing the P2P infrastructure for the PEVs, and also for limiting the PEVs’ interactions to their neighbourhoods.

The P2P network provides two services: (i) information sharing, and (ii) communication. Information sharing service is used to provide agents with all information required for deciding what coalition is better for them, which includes their: energy profile, current coalition, and respective value. Based on the shared information, it can be inferred, for instance, whether a coalition remains feasible if a new agent joins it. On the other hand, the P2P network provides a communication service, which is necessary during the coalition negotiation process.

Along these lines, the P2P network can be seen as a suitable way of providing the required services for our approach. However, it should be noted that the communication layer was modelled to be a seamless interface among the agents. Although we have modelled it as a P2P network, the protocol is not an essential part of our approach. Thus, we no longer focus on P2P technicalities henceforth.

### 4.4 Problem Formulation

The problem can be represented by a graph, where the PEV-agents are expressed by nodes, and the neighbour relations among them are represented by edges. An example is presented in Figure 1. As can be seen, the neighbours of agent 1 are the agents 7 and 9. In this case, $\{1, 7, 9\}$ is a feasible coalition because the agents 1, 7 and 9 are neighbours of each other.

The neighbour relation formulated in Definition 1 is not transitive. In other words, given three agents $a1$, $a2$ and $a3$, if $a1$ is neighbour of $a2$ and $a3$, it does not mean that $a2$ and $a3$ are also neighbours. Considering the example of Figure 1, both agents 8 and 9 are neighbours of agent 3 but there is no neighbour relation between them, so the coalition $\{3, 8, 9\}$ is infeasible. Thus, feasible coalitions are always complete subgraphs.
Our approach is formalised in the form of CFGs, and any kind of characteristic function could be used here. In our case, the characteristic function represents how much the grid is willing to pay beyond the base price for the amount of energy supplied by a coalition. Specifically, the coalition value is a function of the coalition’s total power rating and duration (the greater/longer, the better). This characteristic function slightly differs from the one in [17, 18] since it measures the profit for the total amount of energy rather than per energy unit. In this sense, the value of coalitions can be seen as an incentive for agents to form lasting coalitions, with increased power rating.

The characteristic function is formulated through Equation (2). The value of feasible coalitions is calculated through the first line, and of infeasible coalitions through the second line. Variable $W_C$ is the total power rating of coalition $C$, and is given by Equation (3), where $w_i$ is the power rating of PEV $i \in C$. Variable $T_C$ is the coalition duration, and is measured by the time-slot the coalition is expected to last until, as in Equation (4). The greater the product of $W_C$ and $T_C$, the higher the coalition value. The base price per energy unit is given by parameter $p$. We highlight that energy is measured in kilowatt-hour (kWh)\(^3\). Parameter $\epsilon$ allows the grid to control the maximum value it wants to pay for an energy unit. Finally, parameter $c_0$ is a high-valued normalising constant, which is used to avoid values greater than 1 on term $W_C \times T_C$. In the first line of Equation (2), the second term (within parenthesis) refers to the price per energy unit. The value of a feasible coalition is the product of its power rating and the energy price on it.

$$v(C) = \begin{cases} W_C \times (p \times \min(\frac{W_C \times T_C}{c_0}, \epsilon)) & \text{if } \forall i, j \in C, i \neq j (s(T_i, T_\cdot) \leq \alpha) \\ 0 & \text{otherwise} \end{cases}$$  

(2)

$$W_C = \sum_{i \in C} w_i$$  

(3)

$$T_C = \min_{i \in C} T_i$$  

(4)

\(^3\)By definition, a power rating of 1 kW over one hour produces 1 kWh of energy.
We assume that all PEVs have the same power rating\textsuperscript{4}. In this respect, we can simplify the payoff division in a way that the value of a coalition is equally divided among its members. Although somewhat naïve, this mechanism seems to be a fair scheme. Furthermore, taking the dynamic aspect of the environment into account (coalitions can be created and break at any time), each agent is paid at every time step, considering the amount of energy it has sold to the grid along that specific time step. This time step payoff will be referred to as \textit{instantaneous payoff} hereafter. Thereby, the instantaneous payoff of a given agent \(i\) can be calculated using Equation (5). Finally, note that, according to equations (2) and (5), the greater the coalition’s power rating and availability, the higher the agents’ revenue. In other words, the agents are better off forming coalitions.

\[ P_i = \frac{v(C)}{|C| \times 60} \quad (5) \]

## 5 Self-Adapting Coalition Formation

In this section, we present the Self-Adapting Coalition Formation (SACF), which is a dynamic heuristic-based method for coalition formation on constraint-based scenarios. SACF was designed to run distributed among the agents, i.e., every agent runs an instance of SACF.

The SACF method consists of several procedures, which are performed by each agent. In this section, we present each of these procedures. At the end of this section, a simulation procedure is also presented, which is responsible for calling the other procedures in the proper way.

### 5.1 Main procedure

The main procedure is performed by each agent and is presented in Procedure 1. Roughly speaking, the main procedure allows the agents to negotiate within their neighbourhoods in order to form feasible coalitions. The negotiation process is started by singletons, which are called \textit{requesters}\textsuperscript{5}. The requesters can ask their neighbours (i) to form a new coalition (in the case of single neighbours), or (ii) for permission to join their existing coalitions. In the former scenario, both the requester and its neighbour are singletons, so a new coalition is going to be created. In the latter scenario, the requester wants to join a coalition that already exists. In this case, the join requests received by a coalition are handled by its \textit{leader}\textsuperscript{6}.

The coalition negotiation process is divided into the following four phases:

\textsuperscript{4}The power rating \(w_i\) was set to 3.3 kW for all agents \(i \in A\). This value was adopted because it is similar to the energy transfer rate of some commercial PEVs. However, any other such value could be used here.

\textsuperscript{5}It is important to note that, in the beginning, all agents act as requesters.

\textsuperscript{6}The leader of a coalition does not have any specific characteristics. Instead, it is just a role assumed by an agent to represent the coalition in negotiation processes, thus avoiding redundant negotiations. When the coalition is broken (i.e., one of its members has left), the leader is also responsible for notifying its members about the event. In this approach, the leader of a coalition is the agent with lowest ID. In real situations, such ID could be easily replaced by any other comparable code, such as the vehicles licence plate.
Table 1: Definitions of the procedures’ variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Initial value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i$</td>
<td>id of the agent who is running the procedure</td>
<td>(agent id)</td>
</tr>
<tr>
<td>$N_i$</td>
<td>list of neighbours of agent $i$</td>
<td>${j \in \mathcal{A} \setminus {i} : s(T_i, T_j) \leq \alpha}$</td>
</tr>
<tr>
<td>$C_i$</td>
<td>agent $i$’s coalition</td>
<td>${i}$ (singleton)</td>
</tr>
<tr>
<td>$C'_i$</td>
<td>best coalition found by agent $i$ in phase 1</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>$C''_i$</td>
<td>coalition accepted by agent $i$ in phase 2</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>$M_{i,qs}$</td>
<td>list of “request messages” received by agent $i$</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>$M_{i,pl}$</td>
<td>list of “reply messages” received by agent $i$</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>$M_{i,nf}$</td>
<td>list of “confirmation messages” received by agent $i$</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>$\text{wait}_A$?</td>
<td>whether the agent is waiting for the coalition consolidation or not</td>
<td>$false$</td>
</tr>
<tr>
<td>$\text{chk}_\text{new}$?</td>
<td>whether the agent is trying to join a better coalition</td>
<td>$false$</td>
</tr>
</tbody>
</table>

- Neighbours’ coalitions checking: requesters check for coalitions in their neighbourhoods and send a joining request to the leader of the one with the highest value.

- Requests processing: every agent who has received a request must decide whether the requester(s) can join its coalition or not. If the agent is a leader, then it can directly accept one or more request. Otherwise, if the agent is a singleton, then the negotiation process must advance to the two next phases\(^7\).

- Confirmation: every requester who has received a positive answer (necessarily sent by a singleton) must confirm whether it still intends to form the new coalition.

- Coalition consolidation: where new coalitions are created with the requesters who have confirmed their join intention.

Essentially, the coalition negotiation process takes place in two different occasions. In the first, two singletons negotiate the formation of a new coalition. Consider the illustrative example of Figure 2a, where agent 8 asks to join agent 3’s coalition (phase 1). On phase 2, agent 3 accepts the request, since it is also singleton, and no better options are available. In this case, agent 3 asks agent 8 whether it still wants to form a coalition. After agent 8 has confirmed its intention (on phase 3), agent 3 creates the new coalition $\{3, 8\}$ (phase 4). Now in the second negotiation scenario, the coalition

\(^7\)The rationale behind the second case (request received by a single agent) is that the requester can also have received requests. In this sense, the confirmation is needed to ensure that the requester is still interested in joining the coalition.
Procedure 1: Main procedure performed by the agents

required: variables initialised according to Table 1;

1 while simulation is running do
   // Neighbours’ coalitions checking phase
   2 if \( C_i = \{i\} \) and \( C_i^\prime = \emptyset \) and \( C_i^\prime\prime = \emptyset \) and not wait_A? then
      check neighbours’ coalitions (Procedure 2);
   3 end
   // Requests processing phase
   4 if \( M_{i}^{rq}\neq \emptyset \) and \( C_i^\prime = \emptyset \) and not wait_A? then
      process requests (Procedure 3);
   5 end
   // Confirmation phase
   6 if \( M_{i}^{rp}\neq \emptyset \) and \( M_{i}^{rp}.answer = \text{“Yes”} \) and \( C_i^\prime \neq \emptyset \) then
      confirmation (Procedure 4);
   7 end
   // Coalition consolidation phase
   8 if \( M_{i}^{cnf}\neq \emptyset \) and \( C_i^\prime\prime \neq \emptyset \) and wait_A? then
      process confirmations (Procedure 5);
   9 end
10 end
11
already exists and the single agent asks to join it. Consider the example of Figure 2b. In this case, the two neighbours of agent 1 are in the same coalition. When agent 1 requests to join the coalition (phase 1), agent 7 accepts it immediately (phase 2), since \( \{1, 7, 9\} \) is feasible and has a greater value than \( \{7, 9\} \).

Besides these two situations, agents within coalitions are also allowed to check for better ones. Such a self-adapting mechanism enables agents to improve their revenue whenever possible. In such case, the agent proceeds as if were joining a new coalition. The only difference is that, in the event the agent is admitted to the coalition, it has to cancel its current coalition before joining the new one. The idea behind allowing agents to check for better coalitions is to avoid local optimal solutions. The agents are always allowed to check for better coalitions.

The four negotiation phases are described in detail in the following subsections.

5.2 Neighbours’ coalitions checking phase

In this phase, agents check for existing coalitions in their neighbourhoods. Such a search behaviour is performed as in Procedure 2, where all coalitions (and even other single agents) in agent \( i \)'s neighbourhood are analysed. For each coalition, the procedure checks whether it remains feasible if agent \( i \) joins it. We emphasize that only feasible coalitions are considered here (according to Definition 2). Among the feasible coalitions identified, the best one is selected, i.e., the one where the energy price is
Creating a new coalition

Joining an existing coalition

Figure 2: The negotiation process for coalition formation, where the requester can ask its neighbours (a) to form a new coalition (if the neighbour is also singleton) or (b) for permission to join their existing coalitions.

It is important to note that, when negotiating a coalition, the agents do not care about the value of such coalition, but about the energy price (per unit) on it. The reasoning here is quite simple: as the payoff (value) of a coalition is equally divided among its members, the agent’s payoff is calculated through the energy price on the coalition. Consider, for instance, that two coalitions have the same value, but different number of agents. In this case, on the basis of the problem definition, the coalition with fewer agents would have a higher energy price. In this sense, during the negotiation process, only the energy price is considered.

This procedure can be also performed by agents that are already in a coalition, as long as the new coalition is better than its current one. In this case, a special flag called $chk_{new}$ is activated so as to properly handle this process in next phases.

5.3 Requests processing phase

The requests processing phase is performed as in Procedure 3. In this phase, agents who have received requests must decide whether or not the requesters can join their coalitions. As previously stated, this procedure might be performed either by leader or single agents. This procedure is divided into: requests evaluation (first if block of Procedure 3), and decisions notification (second if block of Procedure 3).

Many cases are handled in this procedure. The most easily solvable ones occur when just one request is received. In this case, if the agent is the leader of the coalition, it just needs to check if the new coalition (with the requester) is feasible, and if the energy price on it is better than of the current coalition. If so, then the coalition is accepted. Handling the request in this way, the performance of the coalition cannot deteriorate. On the other hand, if the agent is a singleton, then it probably have requested to join a coalition also. This way, the agent only accepts the request if the resulting...
Procedure 2: Checking neighbours’ coalitions (phase 1)

1. $NC_i \leftarrow \{C_j : \forall j \in N_i\}$;
2. $\text{foreach } C \subset NC_i$ do
   3. if $C \cup \{i\}$ is not feasible then
      4. $NC_i \leftarrow NC_i \setminus C$;
   5. end
3. end
4. if $NC_i \neq \emptyset$ then
   5. $C'_i \leftarrow \arg \max_{C \subset NC_i} \frac{v(C)}{|C|}$;
   6. if $\frac{v(C'_i)}{|C'_i|} > \frac{v(C_i)}{|C_i|}$ then
      7. // request permission for leader of $C'_i$ to join its coalition
      8. add a join request to $M_{rqs}$, where $j$ is the leader of $C'_i$;
   9. if $C_i = \emptyset$ then
      10. // agent is trying to form a new coalition
      11. $chk_{new} \leftarrow false$;
    12. else
      13. // agent has found a better coalition to join
      14. $chk_{new} \leftarrow true$;
   15. end
16. end
17. end

coalition is the same as the one it has proposed. We call this a mutual request, since two agents are proposing the same coalition to each other. In this case, only the agent with the lowest ID accepts the request of the other, thus becoming the leader of that coalition.

In another scenario, the agent receives multiple requests. In this case, if all requesters are neighbours, then all can be accepted. Otherwise, all permutations of requesters are computed in order to get the subset that most increases the energy price of the coalition. In Procedure 3 we consider that such a computation is performed by the function $best\_feasible$, which trivially computes the best subset of requesters to be accepted.

After all requests were evaluated, the requesters are notified about the decisions made by agent $i$. If agent $i$ is a singleton, then it sends a notification message to every requester (accepting or rejecting their requests) in order to proceed to the next negotiation phases. Otherwise, if agent $i$ is the leader of an existing coalition, then no further negotiation is necessary, and the coalition can be immediately formed.
Procedure 3: Processing requests (phase 2)

1 \( C''_i \leftarrow \emptyset; \)
2 \textbf{if} \( |M_{i}^{rq}s| = 1 \) \textbf{then}
3 \( C'' \leftarrow C_i \cup \{M_{i}^{rq}s\cdot requester\}; \)
4 \textbf{if} \( C_i \neq \{i\} \) \textbf{and} \( C'' \) is feasible and \( \frac{n(C'')}{|C''|} > \frac{n(C_i)}{|C_i|} \) \textbf{then}
5 \( C'' \leftarrow C''; \)
6 \textbf{else if} \( C_i = \{i\} \) \textbf{and} \( C'' = C' \) \textbf{and} \( i < M_{i}^{rq}s\cdot requester \) \textbf{then}
7 \( C'' \leftarrow \{i, M_{i}^{rq}s\cdot requester\}; \)
8 \textbf{end}
9 \textbf{else}
10 \( C''_i \leftarrow C_i \cup \text{best feasible}(M_{i}^{rq}s\cdot requesters); \)
11 \textbf{end}
12 \textbf{if} \( C''_i \neq \emptyset \) \textbf{then}
13 \textbf{if} \( C_i \neq \{i\} \) \textbf{then}
14 \textbf{foreach} \( j \in C''_i \) \textbf{do}
15 \( C_j \leftarrow C''_i; \)
16 \textbf{end}
17 \textbf{else}
18 \texttt{// send an accept to agents in } C''_i
19 \texttt{add an accept message to } M_j^{pl}, \texttt{where } j \texttt{are all agents in } C''_i;
20 \texttt{wait } \texttt{A?} \leftarrow \texttt{true};
21 \textbf{end}
22 \texttt{// reject some requests}
23 \texttt{add a rejection message to } M_j^{pl}, \texttt{where } j \texttt{are all agents in } M_{i}^{rq}s\cdot requesters \setminus C''_i;
24 \textbf{else}
25 \texttt{// reject all requests}
26 \texttt{add a rejection message to } M_j^{pl}, \texttt{where } j \texttt{are all agents in } M_{i}^{rq}s\cdot requesters;
27 \textbf{end}
Procedure 4: Confirmation (phase 3)

1. $\text{accept?} \leftarrow \text{false}$;
2. if $C''_i \neq \emptyset$ then
3.     if $C''_i = M^\text{rpl}_i.coalition$ and $i > M^\text{rpl}_i.sender$ then
4.         $\text{accept?} \leftarrow \text{true}$;
5.     end
6. else
7.     $\text{accept?} \leftarrow \text{true}$;
8. end
9. if $\text{accept?}$ then
   // send a confirmation message
10.     add a confirmation message to $M^{\text{cnf}}_j$, where $j$ is $M^\text{rpl}_i.sender$;
11.     $\text{wait?} \leftarrow \text{true}$;
12.     if $\text{chk.new}$? then
13.         cancel current coalition;
14.     end
15. else
   // send a cancellation message
16.     add a cancellation message to $M^{\text{cnf}}_j$, where $j$ is $M^\text{rpl}_i.sender$;
17. end

5.4 Confirmation phase

In the confirmation phase, requesters who have received positive notifications from the previous phase must confirm whether or not they still intend to form the new coalitions. This phase is performed as in Procedure 4, only by requesters who have asked to form a new coalition with other single agents. Through this procedure, the agent is able to withdraw its request if a better one was received. Specifically, if agent $i$ has received (in the second phase) a request that is better than the one it has proposed (in the first phase), then $C''_i$ will not be empty. In this case, agent $i$ confirms its intention only if the request was mutual, otherwise it refuses it. If agent $i$ has not received a better request, then it simply confirms its interest in forming the initially proposed coalition.

In some cases, the current negotiation, which was started by agent $i$, corresponds to an attempt of moving to a better coalition (if $\text{chk.new}$? is flagged as true). In such scenarios, after agent $i$ has confirmed its join intention, it needs to leave its current coalition, informing the leader about the event. The leader, in turn, breaks the coalition, leaving the remaining agents to negotiate new coalitions.
**Procedure 5:** Processing confirmations (phase 4)

1. `foreach j ∈ M^{cnf}_{i}.sender do`
2. `if agent j has accepted then`
3. `C_j ← C^m_i;`
4. `end`
5. `end`

---

**5.5 Coalition consolidation phase**

Finally, the coalition consolidation phase is performed as in Procedure 5. In this phase, the single agent (who has received the request in the second phase) creates the new coalition with the requesters who have confirmed their interest in forming the coalition with him (in the previous phase).

**5.6 Simulation**

**Procedure 6:** Simulation procedure

1. `while simulation is running do`
   1. `// if some agent has entered or left`
   2. `if set A has changed then`
   3. `foreach new agent i do`
   4. `initialize i’s variables according to Table 1;`
   5. `end`
   6. `update N_i, ∀i ∈ A;`
   7. `end`
2. `run i (Procedure 1), ∀i ∈ A;`
3. `end`

The simulation procedure is formulated as in Procedure 6. This procedure was designed for controlling and setting up the simulation. Furthermore, the simulation procedure formalises the dynamic aspect of the environment, thus allowing agents to enter or leave the simulation at any time. At each iteration, several steps are performed. Firstly, if a new agent is created, then its variables are initialized according to Table 1, as a part of the P2P sign-in process. Secondly, if the set of agents has changed, then the list of neighbours of each agent is updated. Last but not least, all agents are called to run, on their own, one iteration of Procedure 1. It is important to note that the simulation iterations are not equivalent to time steps.

---

[8] It is noteworthy that only the agent with the greatest ID operates this phase.
6 Empirical Evaluation

In this section, we provide an empirical evaluation of SACF’s performance from different perspectives. The main goals include these analyses: (i) how long SACF takes to run, (ii) how far the provided solutions are from optimum, and (iii) how well SACF behaves in a dynamic scenario, namely a smart grid, regarding PEVs’ payoffs, and coalitions’ stability. In order to answer these questions, our approach is evaluated in two different settings:

Closed world In closed world scenarios, no agent can enter or leave the simulation while it is running. The aim here is to compare SACF against IP and IDP in terms of runtime and solution quality. This evaluation is presented in Section 6.1.

Open world Here, agents are allowed to enter and leave the system. In this kind of scenario, we aim at evaluating how SACF can improve PEVs’ performance on V2G sessions. Recall that neither IP nor IDP are suitable for such dynamic scenarios, thus SACF is not compared with them in open world settings. This analysis is presented in Section 6.2.

The main focus of our approach is on open world settings, which are dynamic and intrinsically more complex. In spite of that, the closed world settings are used to allow a comparison of our approach against IP and IDP, which provide optimal solutions but do not work on dynamic scenarios. It is noteworthy, however, that the present approach is not compared against our previous ones [17, 18]. Indeed, such a comparison would not be fair given that the scenario modelling and characteristic function here are different from that of previous works.

The agents’ parameters were defined as follows. The power rating $w_i$ of all PEVs $i \in \mathcal{A}$ was set to 3.3 kW. With respect to the PEVs’ energy profile, its time unit was defined as half-hour periods. Recall that an energy profile here represents how long the PEV is going to be connected to the grid. In this sense, for each PEV $i \in \mathcal{A}$, a value $T_i$ is drawn from a normal distribution $N(\mu, \sigma)$, bounded by $t_{\min}$ and $t_{\max}$, with $\mu = 8$ (4 hours), $\sigma = 4$ (2 hours), $t_{\min} = 1$ (0.5 hour), and $t_{\max} = 16$ (8 hours). Both the PEVs’ power rating and energy profile are set for each agent at instantiation time.

Observe that we do not intend to be realistic with respect to all the parameters’ values employed here. In the case of the PEVs’ energy profile, however, it can be said that sampling it from a normal distribution is a good approximation. In theory, however, these definitions do not affect the results of our approach.

Regarding the parameters of the characteristic function, the following was defined. The normalising constant $c_0$ needs to be high enough to normalise the product of a coalition’s power rating and energy profile, thus it was set to 10,000. Parameter $\epsilon$ was set to 0.9, i.e., the grid would pay up to 90% beyond the base price to a coalition. The base price $p$ of an energy unit was set to $0.50.

Each simulation instance (experiment) is randomly generated based on the above parameters, either for closed and open world settings. All experiments were performed in an Intel(R) Core(TM) i7-2600 3.40GHz PC, with 16GB RAM, running Ubuntu 13.10 64 bits.
6.1 Closed World Performance

In close world settings, experiments were run for a different number of agents \( n = \{10, 11, \ldots, 20\} \). For each \( n \), 30 different scenarios were generated. In order to provide an accurate comparison, both algorithms were tested in exactly the same scenarios. The results are presented in Figure 3. In the plot, each point represents the average runtime over the 30 scenarios, and error bars represent the associated standard deviation.

As can be observed in Figure 3, our approach outperforms IP and IDP in terms of runtime by many orders of magnitude. The average runtime of SACF was almost constant for a varying number of agents. On the other hand, the average runtime of IP and IDP increases exponentially in the number of agents. For example, the average runtime for 20 agents was greater than 24 hours for IP and greater than 7 minutes for IDP, while SACF took less 10 milliseconds, a difference of many orders of magnitude. This difference is due to the fact that, in our approach, agents locally negotiate the formation of feasible coalitions. Consequently, infeasible coalitions are neither considered during the search process. The IP algorithm, in turn, prunes portions of the search space based on estimated upper bounds. As infeasible coalitions have value zero, they do not allow IP to prune the search space. In fact, literature has reported the inability of IP to handle constrained scenarios [25]. Regarding IDP, though it is not sensitive to the distribution of coalition values, it is not able to prune infeasible solutions from the search space.

Now we analyse how far the solutions generated by SACF are from the optimum. The results are plotted in Figure 4, where each bar represents the average solution quality for a different number of agents. Results were normalised in order to show the percentage of optimum achieved by SACF. It is worth noting that the non-normalised bars behave in an ascending monotonic fashion (as a function of the number of agents). Error bars present the standard deviation.

As seen in Figure 4, the results are promising. With a lower running time, SACF
was able to find good solutions in all experiments. The average solution quality over all experiments is approximately 96.5%, with a standard deviation of 5.5%. In almost 90% of the scenarios, SACF has achieved more than 90% of the optimal solution. Furthermore, the optimum was found in almost a half of the scenarios. Such results suggest a trend of SACF to produce good solutions. Based on this, it can be said that SACF is efficient on providing good and fast results, thus being suitable for the dynamic scenarios considered here. It should be noted, however, that SACF cannot be said to always produce good results for scenarios other than those experimented here.

6.2 Open World Performance

In this section, SACF is empirically evaluated in open world settings. In such dynamic scenarios, at any time agents may leave and enter the simulation. Despite abstract, this kind of scenario is equivalent to a real smart grid, where PEVs may enter or leave V2G sessions according to their energy profiles, which in turn are ruled by their owners’ preferences. The idea here is to evaluate how well SACF behaves in such settings, trying to provide an idea of the suitability of our approach for real situations.

For the open world experiments, scenarios were randomly generated with hundred agents. An experiment here consists of a random scenario simulated along 1440 time steps, which corresponds to 24 hours. To keep the number of agents in the simulation stable, whenever an agent leaves the simulation, a new one enters it. Given that the agents’ profiles were modelled in half-hour periods, changes (some agents may leave, and consequently new ones may enter) in the set of agents may occur every half hour. We highlight that such a methodology was defined to provide a clearer analysis of the results, and our approach is not limited in this sense. The remaining parameters were defined according to the definitions at the beginning of this section. Concerning the stochastic nature of the scenarios generation, 30 replications were made in order to
provide more precise results. The results shown hereafter represent the average over these replications, except when otherwise stated.

Figure 5 plots the average coalition structure value ($V(CS)$) along time. In the plot, one can observe long plateaus followed by short, steep valleys. Each of these plateaus represent a half-hour period, and the valleys represent the exit of one or more agents. Such behaviour is due to the fact that, in our experiments, the PEVs’ energy profiles are given in half-hour periods. In this sense, on every half-hour period some agents exit the system (and others enter to replace them). Whenever a PEV leaves the system, its coalition is broken. In such an event, other PEVs of the broken coalition need to negotiate to form new coalitions. The new agents also take part of such negotiation. On spite of the valleys, however, the rapid increases that follow them indicate that the negotiation process is fast enough to recover the value of the previous plateau. Furthermore, among the 48 plateaus, only 16 are strictly below the average coalition structure value, which is approximately $106.80$.

Another interesting observation is that the valleys after plateaus deteriorate the solution in 10%, on average, and in 20%, at most. Considering that several agents exit the system on each half-hour period, we can say that the coalition structure value does not vary significantly from one plateau to another. This is due to fact that, by limiting PEVs to form coalitions with peers of similar energy profile, we prevent the coalitions to break early due to agents with shorter availability than others.

The plot in Figure 6 presents the average and accumulated agents’ payoff along time. Recall that the agents’ payoff is measured by means of Equation (5), which gives the payoff of a given agent $i$ on every time step. As expected, the plateaus-valleys behaviour is also present in this plot. However, only 11 plateaus are strictly below the average payoff along time, which is $0.017$. Here, the deterioration between plateaus is also 10% on average. Thus, not only coalitions, but also the agents’ payoff remain relatively stable along time.
Regarding the agents profit, the average payoff accumulated by a PEV after 8 hours connected to the grid was $8.5. Recall that the payoff represents only the value beyond the base price. In this sense, the total value accumulated by an agent after 8 hours supplying energy to the grid would be, on average, $21.7. On the other hand, singletons have a zero payoff, receiving only $13.2, which refers to the base price. Therefore, agents acting in coalitions received, on average, 65% more than singletons. This represents a key advantage of our approach, meaning that agents have a real incentive to form and remain in coalitions.

### 6.3 Discussion

On the basis of the experiments carried out in this section, SACF can be considered an efficient approach to address the coalition formation problem in dynamic scenarios. The runtime of SACF is almost constant, whereas it grows exponentially with the number of agents in IP and IDP algorithms. In terms of quality, the solutions generated by our method can be said reasonably good, especially when considering the associated runtime. In dynamic environments, SACF has performed satisfactorily in scenarios with a hundred agents. Results also indicate that, thanks to the similarity-based modelling, the coalitions were sufficiently stable along time.

### 7 Conclusions

Coalition formation is an important topic on promoting cooperation in multiagent systems. Among the activities that comprise coalition formation, the coalition structure generation (CSG) task is the one that has drawn more attention of the multiagent systems community. Most existing CSG approaches, despite their achievements, are unsuitable for dynamic, real scenarios.
This work investigates how domain’s constraints might be used in the CSG process in order to prune the solution space. Following this idea, we concentrated on smart grids scenarios, where coalition formation might be beneficial. Specifically, we considered the case of plug-in electric vehicles (PEVs) that are willing to sell their surplus energy on Vehicle-To-Grid (V2G) sessions. It is known that single PEVs cannot operate V2G sessions in a cost-effective way. In this respect, coalition formation has been depicted as a particularly suitable approach to this scenario.

In our approach, an extension to [17] is presented, where PEVs are better off by forming coalitions with peers whose energy profile is similar to theirs. The energy profile of a PEV represents an estimation of how long (in terms of its plug-in and unplug times) it will be available for the V2G session. Along these lines, we presented SACF, a distributed, flexible, and robust method for CSG in dynamic scenarios. In SACF, agents with similar energy profile negotiate among themselves to form coalitions. Here, as opposed to [17], agents are also allowed to check for better coalitions and negotiate their formation, so that local optimal solutions can be avoided. Hence, by limiting the coalitions to agents with similar energy profiles, most of the search space might be pruned, and computational issues faced by traditional approaches are avoided. Another advantage of such a modelling is that coalitions tend to be more stable and to last longer, increasing the PEVs’ profit.

Based on experiments, SACF has shown to be effective in providing reasonably good solutions in a fast, distributed way. Specifically, we showed that our approach outperformed state-of-the-art algorithms in terms of runtime by many orders of magnitude. In terms of quality, the solutions provided by SACF achieved, on average, 96.5% of the optimum (with a standard deviation of 5.5%). Our approach was also tested in a dynamic environment, representing a smart grid scenario. Results have shown that PEVs actually benefit from forming coalitions with SACF, receiving, on average, a profit 65% greater than the case in which PEVs operate individually (as singletons). Furthermore, the formed coalitions have also shown sufficiently stable over time.

Despite promising results, it is important to highlight that SACF has its drawbacks when compared to traditional approaches. The IP algorithm, for instance, is generic, while SACF takes advantage of specific constraints of a given problem. Also, IP returns optimal solutions, while SACF cannot provide quality guarantees. However, we have empirically shown that SACF provides near-optimal solutions. Moreover, regarding the constraints, we have discussed that many domains have constraints that could be easily modelled in SACF. Finally, another key point of our approach refers to its generality regarding the problem constraints. Although a similarity-based constraint have been used here, the model is abstract enough to encompass any other modelling or restriction (e.g., distance), as evidenced in our previous works [17, 18].

For future work, we would like to formally investigate how the search could be improved in order to provide guarantees on the solutions’ quality. This is a fundamental aspect of CSG approaches. Another promising direction would be to extend our modelling to accommodate not only one constraint, but a few. This kind of modelling would be helpful when antagonistic objectives are considered. Also interesting would be to extend the model to handle agents’ uncertainties, e.g., regarding their energy profiles. In such cases, one could impose penalties on coalitions unable to fulfil their commitment to the grid. Additionally, reputation mechanisms could be used in order
to appropriately rank the agents based on their availability predictions.

Acknowledgements

We thank the reviewers for their valuable suggestions. Ramos and Bazzan are partially supported by CNPq and CAPES. Burguillo is supported by the European Regional Development Fund (ERDF) and the Galician Regional Government under agreement for funding the Atlantic Research Center for Information and Communication Technologies (AtlantTIC).

References


Authors’ bios

- Gabriel de O. Ramos received his M.Sc. degree in 2013 from Universidade Federal do Rio Grande do Sul, Brazil. He is currently a Ph.D. student at the same university. His research interests include multiagent systems, multiagent learning, game theory, as well as applications in the domains of traffic & transportation and smart grids.

- Juan C. Burguillo is currently an associate professor at the Department of Telematic Engineering at the University of Vigo. In October 2010, he received a Medal for the Merit Order, from the Ministry of Interior affairs, for supporting the Spanish Security Forces in the fight against childhood crime in the Internet. He has directed and participated in several R&D projects in the areas of Telematics and Computer Science in national and international calls, and has published more than one hundred papers in international refereed journals and conference proceedings. His research interests include intelligent agents, multiagent systems, game theory and evolutionary algorithms applied to optimization.

- Ana L. C. Bazzan received her Ph.D. in 1997 from the University of Karlsruhe, Germany. Currently, she is an associate professor at UFRGS (Computer Science Department) in Porto Alegre, Brazil. She is Associate Editor of the journals Autonomous Agents and Multiagent Systems, Advances in Complex Systems, and Journal of Multiagent and Grid Systems, and served as co-general chair of the AAMAS 2014 conference. Her research interests include multiagent systems, multiagent learning, complex systems, agent-based simulation, and applications of AI and multiagent techniques in traffic simulation and control.