

RESEARCH

Open Access

# Dynamic constrained coalition formation among electric vehicles

Gabriel de O Ramos<sup>1\*</sup>, Juan C Burguillo<sup>2</sup> and Ana LC Bazzan<sup>1</sup>

## Abstract

**Background:** The use of electric vehicles (EVs) and vehicle-to-grid (V2G) technologies have been advocated as an efficient way to reduce the intermittency of renewable energy sources in smart grids. However, operating on V2G sessions in a cost-effective way is not a trivial task for EVs. The formation of coalitions among EVs has been proposed to tackle this problem.

**Methods:** In this paper we introduce Dynamic Constrained Coalition Formation (DCCF), which is a distributed heuristic-based method for constrained coalition structure generation (CSG) in dynamic environments. In our approach, coalitions are formed observing constraints imposed by the grid. To this end, EV agents negotiate the formation of feasible coalitions among themselves.

**Results:** Based on experiments, we show that DCCF is efficient to provide good solutions in a fast way. DCCF provides solutions whose quality approaches 98% of the optimum. In dynamically changing scenarios, DCCF also shows good results, keeping the agents payoff stable along time.

**Conclusions:** Essentially, DCCF's main advantage over traditional CSG algorithms is that its computational effort is very lower. On the other hand, unlike traditional algorithms, DCCF is suitable only for constraint-based problems.

**Keywords:** Artificial intelligence; Game theory; Multiagent systems; Smart grids

## Introduction

Electric power is an essential resource for modern societies. Strategic sectors of economy, such as telecommunications, transportation and industrial activities directly depend on electricity. Electricity networks, or, as called, grids, play a key role on making energy available. Roughly speaking, the electricity network is the infrastructure that connects energy producers to energy consumers.

Despite its importance, electricity grids have evolved very little since they were created. The energy demand, in turn, has grown manifold. In order to meet such a demand, the energy industry has invested mostly in the construction of large power plants. However, such policies resulted in non-redundant grids, which strongly depend on non-renewable, highly polluting energy sources, whose availability is becoming increasingly scarce. As a consequence, such an infrastructure has lost efficiency and

safety. Moreover, the demand for reliable, uninterrupted energy supply increases steadily.

In this scenario, the concept of smart grids emerges. According to the US Department of Energy [1], smart grid is a fully automated electricity network, which intensively monitors and controls every element that composes it, being able to supply energy in an efficient and reliable way. One of the main features of a smart grid is the bidirectional flow of energy and communication between its elements. Along these lines, any element can both supply and consume energy so that, e.g., households with own power generation capacity can sell their surplus production to the grid.

An interesting concept that has emerged in the field of smart grids is the vehicle-to-grid (V2G). Through V2G sessions, electric vehicles (EVs) can provide part of the energy available in their batteries to the grid [2]. Such mechanism is important in situations where the grid relies on intermittent renewable energy sources, such as wind and solar. Thus, the energy stored in the EVs' batteries can be used when supply is not able to meet the demand.

\*Correspondence: goramos@inf.ufrgs.br

<sup>1</sup> Instituto de Informática, Universidade Federal do Rio Grande do Sul, Caixa, Postal 15064, Porto Alegre, Rio Grande do Sul, Brazil

Full list of author information is available at the end of the article

Additionally, EVs can make a profit by selling their energy on V2G sessions.

Although the V2G mechanism has several advantages to the grid, participating in V2G sessions in a cost-effective way is not a trivial task for EVs. According to the authors in [3], to act in V2G sessions, EVs must commit to provide a given amount of energy. However, when operating in an isolated way, EVs are not able to fulfil their commitment due to their insufficient energy capacity and unpredictable availability. In order to address this problem, a particularly interesting approach is the formation of coalitions among EVs, forming virtual power plants (VPPs) [4-7]. By operating in coalitions, EVs can coordinate in order to provide more accurate predictions about their energy availability. Furthermore, the amount of available energy is also increased. Therefore, coalition formation has shown to be an effective way of increasing the profitability of EVs [2,3].

Coalition formation is a research topic that has recently received great attention in the field of multiagent systems. A coalition can be defined as a group of agents that decide to cooperate in order to achieve a common goal. According to the authors in [8], coalition formation includes three activities: coalition structure generation (CSG), solving the optimization problem of each coalition and division of the obtained value among the agents. Among these activities, CSG is the most interesting one for the multiagent systems community.

Traditional CSG algorithms [8-10] deal not only with grouping agents but also grouping them in order to obtain the greatest possible reward. Such activity, however, has been proven to be NP-complete [8]. Moreover, traditional CSG algorithms do not deal with dynamically changing scenarios. In this paper, our focus is not on finding the optimal solution, but on finding a good one (in a reasonable amount of time) in dynamic environments, i.e., environments that change constantly (agents can enter or leave the system). We remark that traditional methods for CSG do not deal with this issue.

In this paper we propose the formation of coalitions among EVs in order to form VPPs. As stated in [2,3], by forming coalitions, EVs can increase their efficiency and, consequently, their profitability. Therefore, we present a distributed heuristic-based method for CSG in dynamic environments, called Dynamic Constrained Coalition Formation (DCCF henceforth), which is able to prevent constrained agents to form a coalition. DCCF was designed to run in a distributed way, in order to be fast and to provide reasonably good solutions. We show that compared to state of the art CSG algorithms, DCCF outperforms them by many orders of magnitude, providing solutions whose average quality nears 98% of the optimal (with a standard deviation of 1.4%) in all tested cases. In more dynamic environments, DCCF also shows good results, being able to keep the agents' payoff sufficiently

stable over time. Also, agents which joined coalitions have obtained a higher profit than isolated agents. In essence, DCCF's main advantage over traditional CSG algorithms is that its computational effort is very low. On the other hand, unlike traditional CSG algorithms, DCCF is suitable only for constraint-based problems.

The remainder of this paper is organized as follows. In 'Background on coalition formation' subsection, we give more details about coalition formation. Related work about coalition formation in smart grids is discussed in 'Coalitions in smart grids' subsection. In 'Problem modelling' and 'Dynamic constrained coalition formation' subsections, we present the problem modelling and our method, respectively. Then in 'Results and discussion' subsection, we present the experiments and analysis of our approach. Conclusions and future work directions are presented in 'Conclusions' section.

### Background on coalition formation

In this section we briefly present the background of coalition formation. The organization of agents in an efficient way is a major challenge in multiagent systems. According to Horling and Lesser [11], different organizational paradigms can be used in order to coordinate agents. Among these, coalition formation stands out. A coalition is a group of agents that cooperate to achieve a common goal, aiming to improve their performance. Specifically, given a set of agents  $A = \{1, 2, \dots, a\}$ , a subset of it is denominated a coalition  $C$ . The partition of the set  $A$  into disjoint and exhaustive coalitions is called a coalition structure (CS) [8]. In the literature, coalition formation is commonly studied in the form of characteristic function games (CFGs). In CFGs, a characteristic function  $v : 2^A \rightarrow \mathbb{R}$  assigns a value  $v(C)$  to each coalition  $C \subseteq A$ . Likewise, the value of a coalition structure CS is given by  $V(CS) = \sum_{C \in CS} v(C)$ .

Regarding characteristic functions, two important concepts are superadditivity and subadditivity [8]. A characteristic function is superadditive if any pair of disjoint coalitions  $C'$  and  $C''$  is better off by merging into one coalition, i.e.,  $v(C' \cup C'') \geq v(C') + v(C'')$ . On the other hand, a characteristic function is subadditive if all agents are better off by operating in an isolated way, i.e.,  $v(C' \cup C'') < v(C') + v(C'')$ . In this paper, the focus is neither in superadditive nor subadditive games, since they represent trivial solutions [8].

As previously stated, coalition formation includes three activities. Among these, CSG has drawn more attention of the multiagent systems community. In CSG, the aim is to find the optimal coalition structure, i.e., the one with the highest value, which is commonly referred as  $CS^*$ . However, there is a scalability issue: the number of possible coalitions is  $2^a - 1$  and of coalition structures is asymptotically in the order of  $O(a^a)$  and  $\omega(a^{\frac{a}{2}})$  [8]. Furthermore,

Sandholm et al. [8] have proved that this problem is NP-complete. Many methods have been proposed in order to solve this problem based on heuristics [12], dynamic programming [10], or in the use of anytime algorithms [9], some of which are discussed next.

Concerning anytime algorithms, Rahwan et al. [9] proposed an integer-partition based algorithm (IP), which uses branch-and-bound techniques. IP is based on an efficient search space representation, where coalitions are grouped by their sizes (in coalition lists) and coalition structures are grouped by the size of coalitions they have (called configurations). IP is divided into three stages: pre-processing (establishes initial bounds on every configuration); choosing the best configuration (bounds are recalculated, the best configuration is selected for the next stage and configurations with low upper bound are pruned); finding the best CS in the chosen configuration. Such approach has to search  $O(a^a)$  coalition structures in the worst case. Recently, we have proposed a pre-processing phase for IP called CPCSG (acronym to constraint-based pruning of coalition structures graph) [13]. In such approach, domain information is used in order to identify infeasible coalitions, which allows the pruning of the search space before the search is started by IP. In the worst case, however, the search space remains  $O(a^a)$ .

With respect to dynamic programming approaches, the state of the art is represented by the improved dynamic programming algorithm (IDP) [10]. The IDP algorithm uses basically the same idea as DP [14], which was originally proposed to solve the set partitioning problem. It works by solving each possible coalition, i.e., deciding whether it is better to split it into two small coalitions or to keep it as it is. Based on this, two tables are kept in memory:  $f_1$  (the solution of each coalition) and  $f_2$  (the value of each solution). After the two tables were filled, the optimal coalition structure can be easily found. In terms of worst case computational complexity, IDP ( $O(3^a)$ ) is better than IP ( $O(a^a)$ ). IP, in turn, is able to return near-optimal solutions anytime. We remark, however, that neither IP nor IDP works on dynamic scenarios.

Other approaches have been proposed to solve the CSG problem from different perspectives. Voice et al. [15] propose an IDP-based algorithm that addresses CSG in graph-constrained settings. Although faster than IDP (under certain conditions), it remains unsuitable for dynamic scenarios. Furthermore, despite being faster than IDP, it still does not scale well for real problems. Coalition formation in graph-constrained scenarios is also addressed in the work of Chalkiadakis et al. [16]. Nonetheless, their work is much more concerned with payoff division than CSG itself. Ueda et al. [17] proposed the use of distributed constraint optimization (DCOP) instances to solve the CSG problem. However, the focus of their

work is not to find optimal CSG solutions (the best CS), but optimal DCOP solutions (the correct value of the coalitions). Chalkiadakis and Boutilier [18] 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. Ramchurn et al. [19] study coalition formation in task-oriented domains. Their approach, nonetheless, is suitable for up to ten agents and does not address CFGs.

Therefore, it is clear that traditional CSG methods are not suitable for dynamic scenarios. To this respect, in this work we propose a heuristic method, which is able to tackle dynamic scenarios. Nevertheless, both IP and IDP address the CSG problem as CFGs and provide optimal solutions in general static cases. Thus, IP and IDP will be used as comparison in our experiments.

#### Coalitions in smart grids

The use of coalitions in smart grids has been widely discussed in the multiagent systems community (see [20] for overview). One of the main interests of the field has been to increase the reliability of renewable energy production.

In [4], Chalkiadakis et al. propose coalition formation among distributed energy resources (DERs) to form VPPs. DERs are renewable energy sources with small-to-medium energy capacity, like wind turbines and solar panels. Taking into account that renewable energy sources are intermittent due to weather conditions, their approach suggests grouping DERs in order to aggregate their production, thus improving their reliability and efficiency. The proposed mechanism incentivizes DERs to provide accurate estimates of their energy production, rewarding good ones. However, this approach has a primary focus on mechanism design rather than on coalition formation, disregarding how far the solution is from the optimal one.

Another approach is the one of Kamboj et al. [5], which proposes the formation of coalitions among EVs in order to operate 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 large 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, considering that vehicles remain parked 96% of the time [2], the use of EVs' batteries would help to reduce costs and to improve efficiency of the regulation market. However, such approach addresses coalition formation in an *ad hoc* fashion, disregarding the solution quality.

In the work of Mihailescu et al. [6], formation of coalitions among producers and consumers is proposed. In their approach, producers who have an increased energy availability are probabilistically selected to coordinate coalitions. Such coordinators are responsible for inviting other producers to join their coalitions. Consumers join the coalitions whose energy profile is more similar to theirs and also based on their proximity. However, their approach neither addresses coalition formation as CFGs nor cares about the solution quality.

The formation of coalitions among producers and consumers is also addressed in [7], specifically, among wind turbines and EVs, also forming VPPs. The goal here is more specific: solve the problem of intermittent power generation of the wind turbines through the use of EVs' batteries, in order to increase the reliability of this kind of energy. To this end, a payment scheme to incentivize EVs to join a VPP of wind turbines was deployed. However, aspects concerning the coalition formation problem are not taken into account.

Other multiagent-based approaches have been proposed in the domain of smart grids. The works of Gerding et al. [21] and Vandael et al. [22] are both concerned with coordinating the EVs' recharging process in order to avoid overloading the electricity network. However, such a problem differs from the V2G one. Moreover, coalition formation is not addressed in these works.

Therefore, it is clear that existing works have primarily focused on applications of smart grids than on coalition formation itself. Although our approach does not necessarily find the optimal solution, it formulates the problem using the CSG formalism, addressing it as a CFG and providing empirical analysis about the solution quality. In this way, we can say that our work lies between traditional CSG and *ad hoc* methods.

## Methods

### Problem modelling

The scenario presented in our work consists in a smart grid where EV agents sell their surplus energy in V2G sessions. As previously discussed, singleton EVs are unable to operate in a cost-effective way [3]. Thus, forming coalitions among EVs represents a suitable approach to solve this issue. In this work, the grid incentivizes the formation of coalitions among EVs through a monetary value, which is proportional to the coalition's power rating, up to certain limits. We assume that the grid is always willing to buy the energy offered by EVs, i.e., whenever an EV has energy to be sold, the grid will buy it.

Regarding the dynamic aspect of the problem, EVs can enter or leave the system at any time. This is not just a modelling definition but a real aspect of the domain. Actually, the EVs' permanency in the grid is ruled by their owners' preferences. Specifically, EVs should ensure

a minimum energy reserve so as to meet their owners' demand. Thereby, we simplify such a requirement, assuming that EV agents know when to stop selling energy to the grid. At this point, EVs are allowed to leave the system as soon as they deem necessary. Although it might be argued that this behaviour would invalidate the EVs' commitment with the grid, such aspects are out of the scope of this work. Here, the focus is only on finding a near-optimal CSs. Thus, this issue could be better explored in a future work.

A further assumption must be made. As discussed in [13], EVs should supply energy only to consumers who are in the same region as them (or just close enough). Such a constraint exists because power lines have a limited energy flow capacity. Considering that multiple power lines may be used in order to supply energy to a single consumer, travelling long distances may impose a huge burden on the distribution network. Therefore, the distance<sup>a</sup> among the EVs is a constraint that must be taken into account while the coalitions are being formed. Specifically, EVs must form coalitions only with EVs that are close enough. A coalition that fits into such criterion is said a *feasible coalition*. More formally, Equation 1 holds for all feasible coalitions, where  $C$  is a coalition,  $i$  and  $j$  are agents of coalition  $C$ ,  $d_j^i$  is the distance between agents  $i$  and  $j$ , and  $\alpha$  is the maximum distance that is allowed between the agents of a given coalition. On this basis, Definition 1 can be formulated. Importantly, the maximum distance  $\alpha$  must be defined by the grid itself, in order to better represent the capacity of its power lines.

$$\forall i \in C, \forall j \in C \setminus \{i\} (d_j^i \leq \alpha) \quad (1)$$

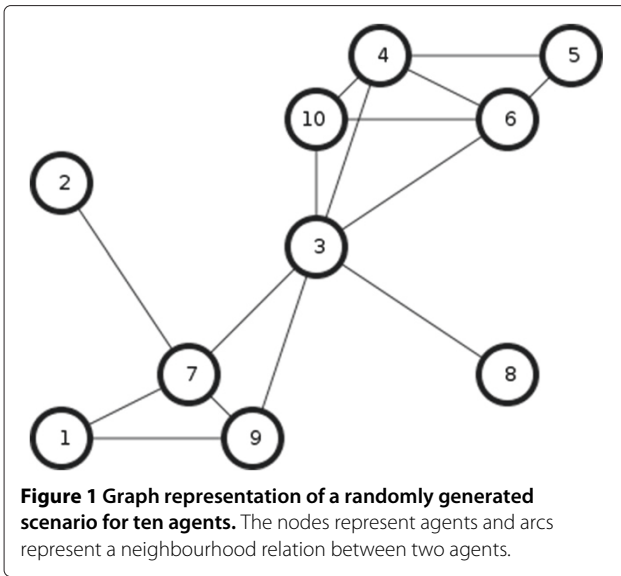
**Definition (Feasible coalition).** A *feasible coalition* is one for which Equation 1 holds.

Another important definition is the one of *neighbours*. In the context of this work, two agents are neighbours if the distance between them is lower than  $\alpha$ , as formulated in Definition 2.

**Definition (Neighbours).** The *neighbours* of a given agent  $i \in A$  are all agents  $j \in A \setminus \{i\}$  for whom  $d_j^i$  is smaller than  $\alpha$ .

The problem can be represented by a graph, where the agents are expressed by nodes and the neighbourhood relation among them is represented by edges. An example is presented in Figure 1. As 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.

Based on the problem proposed in this work, our focus is on EVs that are willing to participate on V2G sessions



only. The proposed approach works as follows. Whenever an EV is plugged into the smart grid, it automatically signs in a peer-to-peer (P2P) network (see next paragraph). Through the P2P network, EVs share personal information with their neighbours (such as location, current coalition and so on). Based on the shared information, the agents can look for feasible coalitions within their neighbourhoods, proposing the creation of the most valuable one, which will increase its profit. Coalitions last until an agent decides to leave it. Agents, in turn, remain in their coalitions until a better one is proposed or until they are plugged into the grid. We go further on each of these steps in next subsection.

Concerning the P2P network, it is important to explain its role in this work. Basically, the P2P network consists of one hub and many leaves connected to it, similarly to the Gnutella2 protocol. The topology of the used P2P network is analogous to the proposed smart grid scenario. Hubs are substations and leaves are EVs. A substation controls the portion of the distribution network where the EVs are in. In this way, the information shared by the agents on the P2P network are their location, whether they are in a coalition or not, and the value of their current coalition. Through this information, an agent can improve its performance, proposing only those coalitions that are most likely to be accepted by its neighbours. As will be defined in Definition 3, such coalitions are referred as *potential coalitions*.

The P2P network is also used for communication purposes, i.e., message exchange. Thus, the P2P network can be seen as a communication layer, through which the agents can share information and even communicate among themselves. It is important to note that in order to make the communication effective, the hub ensures that

the agents can reach their neighbours only, instead of all agents. Therefore, the P2P network shows to be a suitable way to ensure the communication among the agents. It is noteworthy that the communication layer was modelled to be a seamless interface among the agents. Although we have modelled it like a P2P network, the protocol is not an essential part of our approach. Thus, hereafter, we no longer focus on P2P particularities.

Our approach can be formalized in the form of CFGs. The value  $v(C)$  of a given coalition  $C$  represents how much the grid is willing to pay beyond the normal price for each energy unit sold by the agents in that coalition. The coalition value is a function of the coalition's total power rating (the greater, the better). The total power rating of a coalition  $C$  is given by Equation 2, where  $w_i$  is the power rating of EV  $i$ . Thus, the value of coalitions can be seen as an incentive for agents to join coalitions.

$$W_C = \sum_{i \in C} w_i \quad (2)$$

In this work only one kind of agent was defined, the EV. The aim of the EV agents is to sell the surplus energy on their batteries to the grid, getting the highest profit (pay-off) possible. It is worth noting that in our approach,  $v(C)$  does not represent the payoff that is going to be divided among all members of coalition  $C$ . Instead, it represents how much incentive the agents of coalition  $C$  receive for each energy unit sold by them. In this sense, the payoff obtained by a given agent  $i \in C$  is the product of its power rating  $w_i$  and the value  $v(C)$  of its current coalition. Thus, we can reformulate the agents' objective as joining the coalitions which have the highest values. Consequently, the agents act selfishly, looking for the coalitions where their energy will be more valuable. Along these lines, the definition of a potential coalition can be completed through Definition 3.

**Definition (Potential coalition).** A *potential coalition* is one which is feasible and  $\forall i \in C (v(C) > v(C_i))$ , where  $C_i$  is the current coalition of agent  $i$ .

As previously stated, the environment can be dynamic, i.e., coalitions can be formed and terminated at any time. In this sense, our model simplifies the agents' payoff division in a way that each agent is paid on every time step<sup>b</sup> based on the amount of energy it has sold to the grid during that time step. This time-step payoff will be referred to as *instantaneous payoff* hereafter. We highlight that energy is measured in kilowatt hour (kWh)<sup>c</sup>. Therefore, the instantaneous payoff of agent  $i$  can be obtained through Equation 3:

$$P_i = \frac{w_i * v(C)}{60}. \quad (3)$$

### Dynamic constrained coalition formation

Following the modelling discussed in the previous section, we present DCCF, which is a distributed heuristic-based method for constrained CSG in dynamic environments. DCCF was designed to run distributed among the agents, i.e., every agent in the environment runs an instance of DCCF.

The DCCF method consists of several procedures. The main procedure is detailed in the next subsection. The other procedures, which represent the coalition negotiation phases, are detailed in next subsections. Finally, an illustrative example is presented in the last subsection.

#### Main procedure

DCCF's main procedure has two parts. The first one, which we will refer to as the simulation procedure (Algorithm 1), was designed for controlling and setting up the simulation. The simulation procedure formalizes the dynamic aspect of the environment, making possible that agents enter or leave the simulation at any time. On each iteration, several steps are performed. First, if a new agent is created, its variables are initialized according to Table 1 (lines 3 to 5 of Algorithm 1), as a part of the P2P sign-in process. Second, the agents' lists of neighbours are updated every iteration (line 6), if required. Last but not the least, all agents are called to run, on their own, one iteration (line 8) of Algorithm 2. It is important to note that iterations are not equivalent to time steps.

The second part of main procedure refers to the agents' execution and is presented in Algorithm 2. This procedure is performed by each agent on each iteration of the simulation. Basically, it allows the agents to negotiate among themselves to form potential coalitions. The negotiation process takes place through information exchange among the agents. Specifically, every agent is able to find potential

---

#### Algorithm 1: Simulation procedure

---

```

1 while simulation is running do
    // 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  $N_i, \forall i \in A$ ;
7     end
8   run  $i$  (Algorithm 2),  $\forall i \in A$ ;
9 end

```

---



---

#### Algorithm 2: Agents' execution procedure

---

```

// Neighbours invitation phase
1 if  $C_i = \{i\}$  and  $C'_i = \emptyset$  and  $C'' = \emptyset$  then
2   | invite neighbours (Algorithm 3);
3 end

// Invitations processing phase
4 if  $M_i^{inv} \neq \emptyset$  then
5   | process invitations (Algorithm 5);
6 end

// Replies processing phase
7 if  $M_i^{rpl} \neq \emptyset$  and  $C_i = \{i\}$  and  $C'_i \neq \emptyset$  and  $C'' = \emptyset$  then
8   | process replies (Algorithm 6);
9 end

```

---

coalitions (based on information shared by its neighbours) and negotiate their formation (through message exchange).

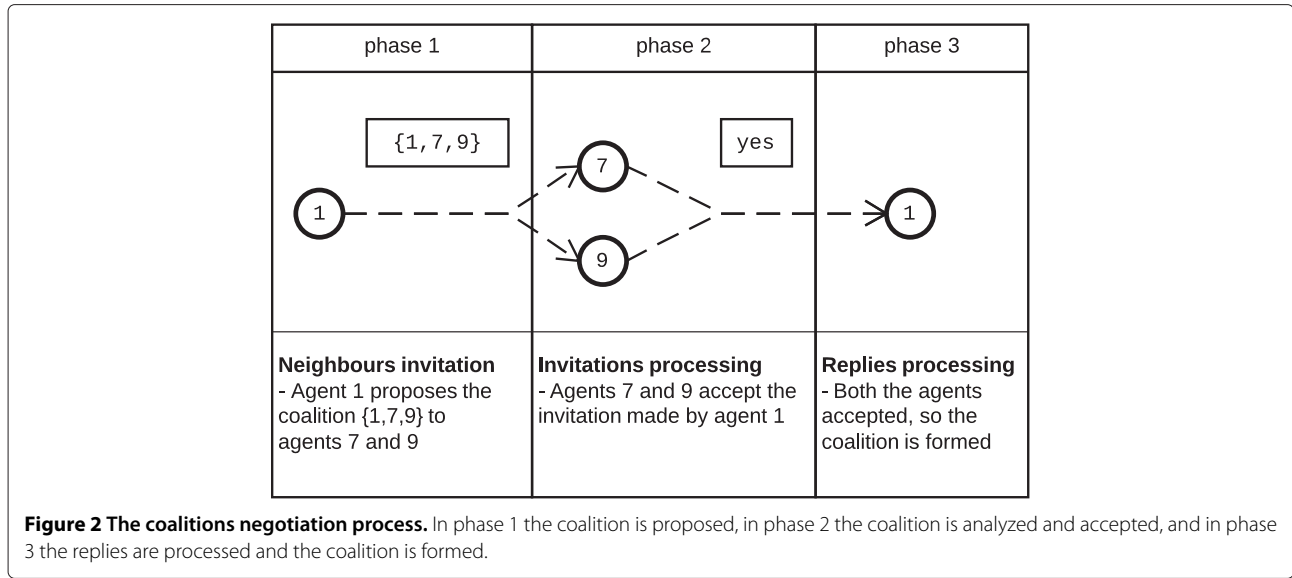
The coalition negotiation process is divided into three phases:

- Neighbours invitation (lines 1 to 3 of Algorithm 2): singleton agents propose potential coalitions to their neighbours
- Invitations processing (lines 4 to 6 of Algorithm 2): agents who have received invitations choose and accept the best one
- Replies processing (lines 7 to 9 of Algorithm 2): agents who proposed coalitions process the received replies and form (or not) the coalitions

An illustrative example of these phases is presented in Figure 2. In the first phase, agent 1 proposes to its neighbours the formation of coalition  $\{1, 7, 9\}$ . In the second phase, agents 7 and 9 evaluate the invitation and accept it.

**Table 1** Definitions of the procedures' variables

Variable	Description	Initial value
$i$	ID of the agent who is running the procedure	(Agent ID)
$N_i$	List of neighbours of agent $i$	$\{j \in A \setminus \{i\} : d_j^i \leq \alpha\}$
$C_i$	Agent $i$ 's coalition	$\{i\}$ (singleton)
$C'_i$	Coalition proposed by agent $i$ in phase 1	$\emptyset$
$C''_i$	Coalition accepted by agent $i$ in phase 2	$\emptyset$
$M_i^{inv}$	List of 'invitation messages' received by agent $i$	$\emptyset$
$M_i^{rpl}$	List of 'reply messages' received by agent $i$	$\emptyset$



Finally, in the third phase, agent 1 processes the received replies and forms the initially proposed coalition.

The coalition negotiation phases are described in detail in the following sections.

#### Neighbours invitation phase

The neighbours invitation phase, which is performed by every agent  $i$  that is not in a coalition, is structured as in Algorithm 3. The rationale behind this phase is simple: (i) search for feasible coalitions, (ii) sort out those that are potential and (iii) invite neighbours to form the best one. This phase is structured as in Algorithm 3.

---

#### Algorithm 3: Inviting neighbours

---

```

1  $\mathcal{F}_i \leftarrow$  Algorithm 4; // feasible coalitions
2  $\mathcal{P}_i \leftarrow \emptyset$ ; // potential coalitions
3 foreach  $C \subset \mathcal{F}_i$  do
4   foreach  $j \in C$  do
5     if  $C_j \neq \{j\}$  and  $v(C_j) \geq v(C)$  then
6        $C \leftarrow C \setminus \{j\}$ ;
7     end
8   end
9   if  $C \neq \emptyset$  then
10     $\mathcal{P}_i \leftarrow \mathcal{P}_i \cup \{C\}$ ;
11  end
12 end
13  $C'_i \leftarrow \arg \max_{C \in \mathcal{P}_i} v(C)$ ;
    // send an invitation message to
    agents in  $C'_i$ 
14 update  $M_j^{inv}$  with  $C'_i, \forall j \in C'_i$ ;
```

---

Firstly, Algorithm 4 is used to find feasible coalitions, by means of Definition 1, among  $i$ 's neighbours. Second, potential coalitions are selected among the feasible ones, based on Definition 3. Finally, the best potential coalition (i.e., the one that maximizes the agents payoff) might be selected and proposed by  $i$  to its neighbours.

#### Invitations processing phase

In the invitations processing phase, agents must reason about the best invitation to accept. The best invitation received is the one that proposes the coalition with highest value. This procedure is presented in Algorithm 5. Roughly speaking, the best invitation can be accepted if

---

#### Algorithm 4: Finding feasible coalitions

---

```

1  $\mathcal{F}_i \leftarrow \emptyset$ ;
2 foreach  $j \in N_i$  do
3    $\mathcal{F}_i \leftarrow \mathcal{F}_i \cup \{\{i, j\}\}$ ;
4   foreach  $k \in N_j, k > j$  and  $k \in N_i$  do
5     foreach  $C \subset \mathcal{F}_i$  do
6       if  $C \setminus N_k = \emptyset$  then
7          $C \leftarrow C \cup \{k\}$ ;
8       else
9          $C_{new} \leftarrow (C \setminus (C \setminus N_k)) \cup \{k\}$ ;
10       $\mathcal{F}_i \leftarrow \mathcal{F}_i \cup \{C_{new}\}$ ;
11    end
12  end
13 end
14 end
15 return  $\mathcal{F}_i$ ;
```

---

---

**Algorithm 5:** Processing invitations

---

```

1  $\hat{m} \leftarrow \arg \max_{m \in M_i^{inv}} v(m.coalition);$ 
2  $accept? \leftarrow false;$ 
3 if  $C'_i \neq \emptyset$  then
4   if  $\hat{m}.coalition = C'_i$  and  $\hat{m}.sender < i$  then
5      $accept? \leftarrow true;$ 
6   end
7 else if  $v(\hat{m}.coalition) > v(C_i)$  then
8    $accept? \leftarrow true;$ 
9 end
10 if  $accept?$  then
11    $C'' \leftarrow \hat{m}.coalition;$ 
12   // accept invitation  $\hat{m}$ 
13   update  $M_{\hat{m}.sender}^{rpl};$ 
14   // reject remaining invitations
15   update  $M_{m.sender}^{rpl}, \forall m \in M_i^{inv} \setminus \{\hat{m}\};$ 
16 else
17   // reject all invitations
18   update  $M_{m.sender}^{rpl}, \forall m \in M_i^{inv};$ 
19 end

```

---

the corresponding coalition's value is higher than one of  $i$ 's current coalition (lines 7 to 9).

One additional case must be handled. If agent  $i$  has already proposed a coalition to its neighbours and, coincidentally, the best invitation received is to form the same coalition it has proposed. This case is called a *mutual invitation*, since two (or more) agents are inviting each other for the same coalition. To address this case, all agents reply the invitation only to the agent with lowest ID<sup>d</sup> (as in lines 3 to 6). This way, only the agent with lowest ID will perform the third phase.

Finally, if a coalition has been accepted, then the agent sends a message to the neighbour who has made the invitation, notifying it about its choice (line 12). All non-accepted invitations are rejected.

**Replies processing phase**

Finally, the replies processing phase is performed as in Algorithm 6. Essentially, the coalition can be formed only if all replies are positive, i.e., if all agents of the proposed coalition have accepted the invitation. If there is a negative reply, then the proposed coalition does not form.

In the event that the coalition is formed, the members' P2P-shared information is updated. In order to simplify the procedure, in this paper we entrusted the agent who proposed (agent  $i$ ) the coalition with this task (as lines 4 to 9). Additionally, agents who were already in coalitions must leave them to enter the new one (lines 5 to 7).

Importantly, agents who receive a *cancellation* or a *coalition finished* message simply return to their initial state, being able to perform phases 1 and 2 again.

---

**Algorithm 6:** Processing replies

---

```

1 if  $\exists \hat{m} \in M_i^{rpl} (\hat{m}.answer = "No")$  then
2   send a cancellation message to every agent  $i \in C'_i;$ 
3 else
4   foreach  $j \in C'_i$  do
5     if  $C_j \neq \emptyset$  then
6       leave  $C_j$  (notify members of  $C_j$ );
7     end
8      $C_j \leftarrow C'_i;$ 
9   end
10 end

```

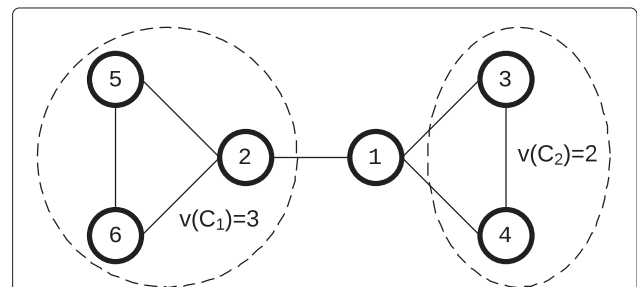
---

**Illustrative example**

In order to explain how DCCF works, we describe the following example.

Assuming that the scenario is as in Figure 3, where there are six agents. One of these agents is isolated (agent 1), and the others are organized into two distinct coalitions:  $C_1 = \{2, 5, 6\}$  and  $C_2 = \{3, 4\}$ . For simplicity, let us assume in this example that the value of a coalition is a function of its cardinality, i.e.,  $v(C_1) = 3$  and  $v(C_2) = 2$  and that individual agents value 0. Now suppose that agent 1 is new in the system and that its neighbours are agents 2, 3 and 4.

After agent 1 has entered, the neighbours invitation phase is started by agent 1. First, it identifies the feasible coalitions:  $\{1, 2\}$  and  $\{1, 3, 4\}$ . By analyzing the feasible coalition  $\{1, 2\}$ , agent 1 realizes that its value ( $v(\{1, 2\}) = 2$ ) is lower than the value of agent 2's current coalition ( $v(\{2, 5, 6\}) = 3$ ). Thus,  $\{1, 2\}$  is not a potential coalition and can be discarded. The second feasible coalition,  $\{1, 3, 4\}$ , in turn, has higher value ( $v(\{1, 3, 4\}) = 3$ ) than of the current coalition of agents 3



**Figure 3** Graph of an example scenario. The scenario consists of six agents, where there are two coalitions (represented by the dashed lines), namely,  $C_1 = \{2, 5, 6\}$  and  $C_2 = \{3, 4\}$  (whose values are  $v(C_1) = 3$  and  $v(C_2) = 2$ ).



and 4 ( $v(\{3, 4\}) = 2$ ). Based on that,  $\{1, 3, 4\}$  is a potential coalition. Therefore, agent 1 invites agents 3 and 4 to form the coalition  $\{1, 3, 4\}$ .

In the second iteration, agents 3 and 4 analyze the invitation made by agent 1. As the value of the proposed coalition is better than their current coalition, both decide to join the coalition.

In the third iteration, agent 1 starts the replies processing phase. As agents 3 and 4 have accepted the invitation, then the coalition can be created. Thus, the coalition  $\{1, 3, 4\}$  is created on iteration 3.

## Results and discussion

In this section we compare DCCF against traditional CSG approaches. We also evaluate DCCF's performance in dynamic environments.

### Value assignment

In order to evaluate our approach, some issues must be detailed. Initially, since this work focuses on characteristic function games, it is essential to discuss how the characteristic function was modelled. The characteristic function is formulated through Equation 4, where  $\delta$  is the expected power rating of any coalition, and  $\epsilon$  defines the maximum financial incentive the grid is willing to pay for a given coalition,  $W_C$  is the power rating of coalition  $C$  has available, and  $p$  is the normal price of an energy unit by

$$v(C) = \begin{cases} 0, & \text{if } \exists i \in C, \exists j \in C \setminus \{i\} (d_j^i > \alpha) \\ \min\left\{\left(\frac{W_C}{\delta}\right)^2 \times \epsilon, \epsilon\right\} \times p, & \text{otherwise.} \end{cases} \quad (4)$$

Through the first line of Equation 4, the values of infeasible coalitions are set to zero. The second line, in turn, assigns the value of feasible coalitions. This takes into account their power ratings<sup>e</sup>  $W_C$  (the greater  $W_C$ , the greater the coalition value). The  $\epsilon$  is used to allow the grid to control the maximum value it wants to pay for an energy unit. Finally, through  $\delta$  the grid defines the desired granularity of coalitions, i.e., the power rating the grid desires each coalition has.

Regarding the values set to the parameters of the characteristic function, the following was defined. Parameter  $\epsilon$  was set to 0.9, i.e., the grid would pay up to 90% beyond the normal price to a coalition. We assume that the grid would like to form small VPPs, whose power rating is around 150 kW. On this basis,  $\delta$  was set to 150. Finally, the normal price of an energy unit,  $p$ , was set to R\$0.50 (approximately the energy price per kilowatt hour in Brazil, in Brazilian currency).

### Types of experiments

In order to evaluate our approach, two sets of experiments were made: in closed and in open world. In the closed-world scenarios, no agents can enter or leave the

simulation after it has been started. The open-world scenarios, in turn, allow new agents to enter the simulation, and existing agents to leave the simulation. The focus of our approach is on open-world scenarios, which are dynamic and more complex. However, in order to compare our approach against other CSG algorithms (which do not work on dynamic scenarios), the use of closed-world scenarios is more suitable.

The agents were randomly positioned in a grid-based scenario. Edges were created between pairs of agents whose Euclidian distance was lower than  $\alpha$ . For all scenarios tested, both in open and closed world settings, the parameter  $\alpha$  was set to 7. The distance here is measured by cells. Indeed, one can imagine that each cell has  $10 \times 10$  m, as adopted in [13].

The set of experiments was performed in an Intel(R) Core(TM) i7-2600 3.40 GHz PC, with 16 GB RAM and Ubuntu 12.04 64 bits.

### Closed-world settings

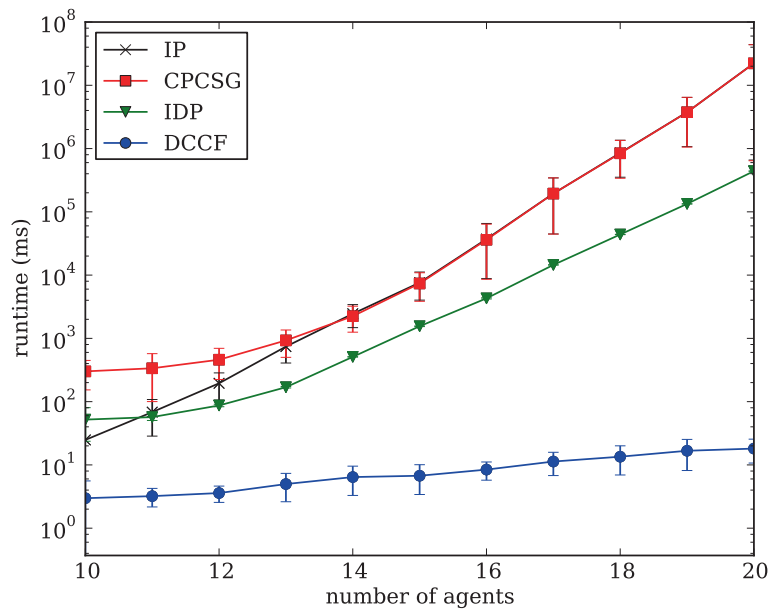
In closed-world settings, DCCF is compared against other CSG approaches: IDP [10], IP [9] and CPCSG [13] in terms of runtime and solution quality<sup>f</sup>.

Experiments were run for different number of agents  $a = \{10, 11, \dots, 20\}$ . For each number of agents, 30 different scenarios were generated (as described in the 'Value assignment' subsection). In order to accurately compare the algorithms, each of them was tested in exactly the same scenarios. Results are presented in Figure 4. In the graph, each point shows the average runtime of the 30 scenarios and the error bars represent the standard deviation.

As can be observed in Figure 4, DCCF outperforms the other algorithms in terms of runtime by many orders of magnitude. While the average runtime of DCCF was lower than 1 s in all sets of experiments, in other algorithms the average runtime increases exponentially with the number of agents. For the sake of comparison, for 20 agents, the IP algorithm takes on average about 6 h to run. DCCF, in turn, takes less than 20 ms.

Now, we analyze how far the solution generated by DCCF is from the optimum. Results are plotted in Figure 5, where the DCCF points show how far the average solutions are from the optimal ones. Results were normalized in order to show the percentage achieved by DCCF in relation to the optimum. Error bars plot the standard deviation of each set of experiments. It is important to note that the non-normalized curves behave in an ascending monotonic fashion (as a function of the number of agents).

As shown in Figure 5, the results are very promising. Although DCCF has taken less than 1 s to run, it was able to find good solutions. In almost all tested cases, the solutions generated by DCCF achieved more than 95% of the optimal solution. The average quality achieved



**Figure 4** Comparison of DCCF against IDP, IP and CPCSG. The runtime (in log scale) for different number of agents (from 10 to 20 agents) is shown.

was approximately 98.1% (averaged over all experiments). Also, the standard deviation was up to 1.4% in all experiments, showing a tendency on producing good solutions. Therefore, DCCF's advantage over the other algorithms is that it runs on a small amount of time, in dynamic distributed environments, achieving good solutions in all tested cases. Importantly, however, DCCF cannot be said to always produce the same results for scenarios other than those experimented here.

#### Open-world settings

In this section, we empirically evaluate DCCF in an open-world setting. To this end, we have generated an initial

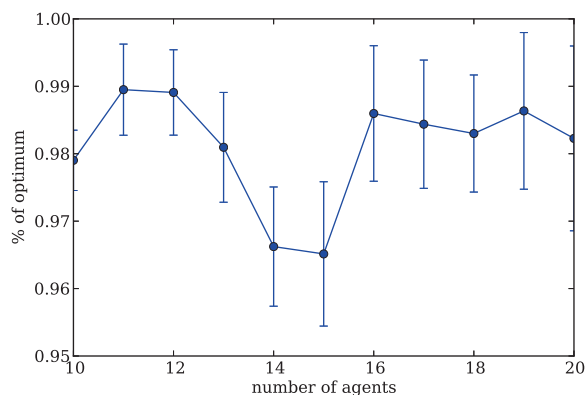
scenario with 40 agents (as described in the 'Value assignment' subsection), which is used in the experiments along this section (Figure 6).

Considering that the world is open, agents can enter and leave in the simulation while it is running. The frequency that such events occur is defined in terms of probabilities. Specifically, the probability of a new agent entering in the system in a given time step is defined by  $\mathcal{P}_e$ . In the same way, the probability of a given agent (selected uniformly random) leaving the system in a given time step is defined by  $\mathcal{P}_l$ . Based on these parameters, on every time step a new agent is created with probability  $\mathcal{P}_e$ ; and with probability  $\mathcal{P}_l$ , a randomly selected agent is eliminated. In the experiments, we have simulated a 24-h period, which corresponds to 1,440 time steps.

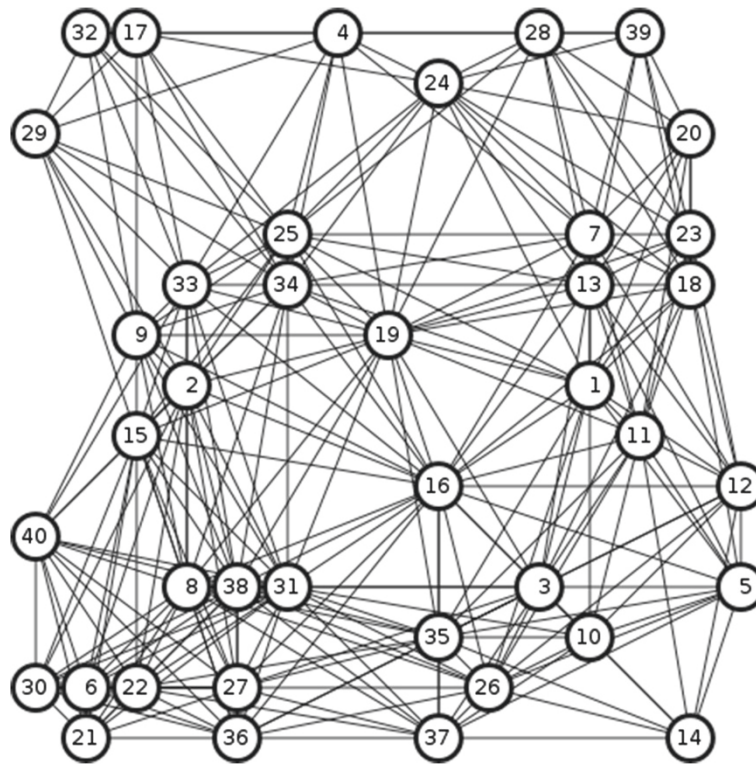
In order to evaluate how DCCF behaves in different conditions, three different settings are tested, all of them using the same 40 agents' initial scenario. In all cases, small values are set to parameters  $\mathcal{P}_e$  and  $\mathcal{P}_l$ , in order to avoid fast changes in the environment. The three settings are defined as follows:

- Setting 1,  $\mathcal{P}_e = \mathcal{P}_l = 0.005$
- Setting 2,  $\mathcal{P}_e = 0.02$  and  $\mathcal{P}_l = 0.005$
- Setting 3,  $\mathcal{P}_e = 0.005$  and  $\mathcal{P}_l = 0.02$

In the first setting, both  $\mathcal{P}_e$  and  $\mathcal{P}_l$  have the same value (0.005). In this sense, the number of agents tends to remain the same along time. The variation in the number of agents over time is shown in Figure 7. Indeed, as can be observed, the number of agents remains stable. In this



**Figure 5** Quality of solution generated by DCCF. The percentage of DCCF in comparison with the optimal solution (normalized), for different number of agents (from 10 to 20 agents), is shown.



**Figure 6** Initial scenario with 40 agents used in the open-world experiments.

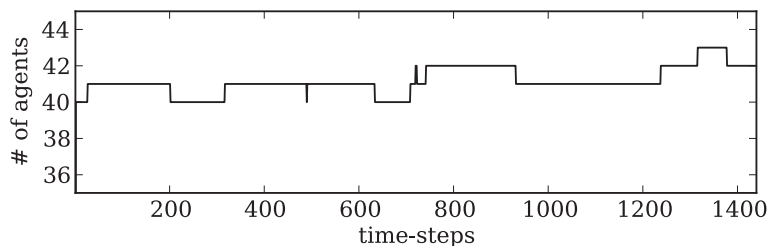
case, the number of agents in the end of the simulation was 42.

Concerning the value of the solutions, Figure 8 plots the variation in the social welfare (the coalition structure value  $V(CS)$ ) over time. The social welfare does not experiment large variations. The greatest variations occur when an agent leaves the system, as it can be observed by comparing the graphs of Figures 7 and 8. Such behaviour shows that DCCF is effective on organizing the agents into coalitions.

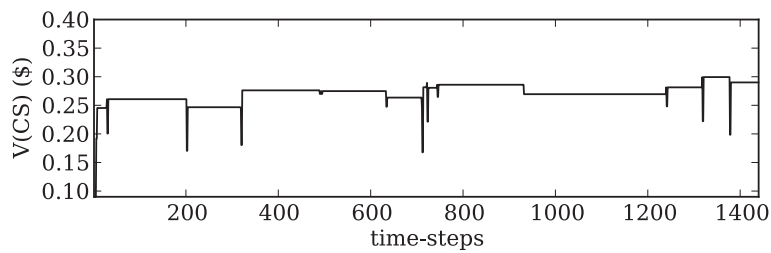
In order to better understand such behaviour, we now show the variation in the agents' instantaneous payoff along time (Figure 9). Recall that the instantaneous payoff is obtained as in Equation 3. The average instantaneous payoff also remains stable over time. Here it is important

to note that the instantaneous payoff is really low (less than R\$0.01). However, this is the received value in just 1 min. Moreover, the payoff considered in this CFG is the value the grid pays beyond the normal price (as described in the 'Value assignment' subsection). In this sense, after the entire 24-h simulation, the average obtained payoff was approximately R\$4.47 per agent. Also, considering the total profit (normal price + coalition price) obtained by the agents, the average was of R\$44.07. In this respect, an agent that remained isolated throughout all simulation has received only R\$39.6. Therefore, agents that were in coalitions have received, on average, a profit 11.28% greater than singleton agents.

Concerning the system stability, another important point is the amount of time the coalitions last. Figure 10



**Figure 7** Number of agents along time, for setting 1.



**Figure 8** Variation of  $V(CS)$  over time, for setting 1.

shows the instantaneous average duration of the coalitions. Whenever an agent enters or leaves the system, new coalitions are created. As a consequence, the curve drops significantly. The average duration of the coalitions throughout the whole simulation was 384 time steps.

To conclude the experiments with the first setting, Figure 11 shows the number of messages<sup>8</sup> exchanged along time. As expected, the amount of exchanged messages increases whenever an agent enters or leaves the system.

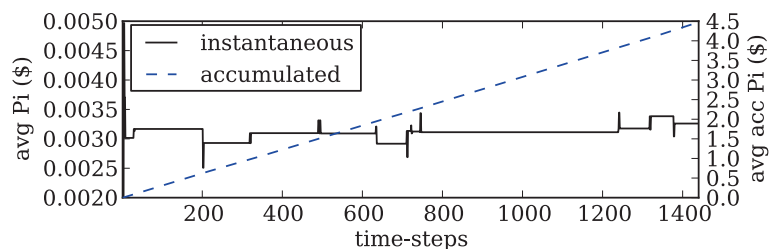
Now we analyze the second setting, where  $\mathcal{P}_1$  remains with the same value as in setting 1, but  $\mathcal{P}_e$  is increased to 0.02, i.e., four times greater than  $\mathcal{P}_1$ . In this way, it is expected that the number of agents increases over time. Indeed, this is what happens, as Figure 12 shows. In this case, the simulation, which started with 40 agents, ended up with 62 agents.

Concerning the agents' payoff, Figure 13 presents the average instantaneous payoff of the agents along time. The behaviour of the instantaneous payoff curve is somewhat less stable than in the case of setting 1. Obviously, this occurs because, in setting 2, the entering of new agents in the system is much more frequent. Consequently, there are more coalitions being proposed and being accepted. However, it can be noted that the payoff increases as well. For the sake of comparison, in setting 1 the average payoff obtained by the agents throughout all simulation was approximately R\$4.47. In the case of setting 2, this value is increased to R\$7.72. This means that agents

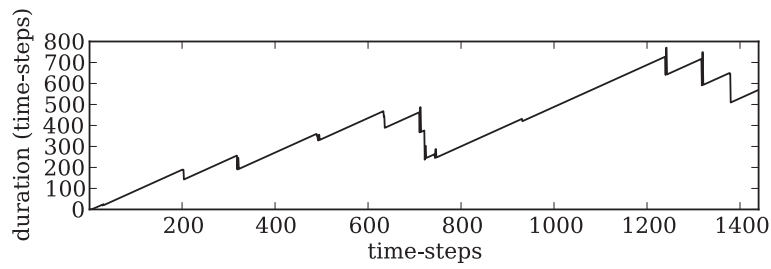
who have joined coalitions obtained a 19.5% greater total profit than singleton agents on average. Such behaviour shows that value of the coalitions increase as they become greater.

The other metrics analyzed in the case of setting 1 have the same behaviour for setting 2. Thus, we do not repeat the plots. Rather, we go straight to setting 3, where  $\mathcal{P}_e = 0.005$  and  $\mathcal{P}_1 = 0.02$ . In this way, the number of agents over time is expected to decrease. These results are depicted in Figure 14. In the case of setting 3, the final number of agents (in the end of simulation) has dropped by half.

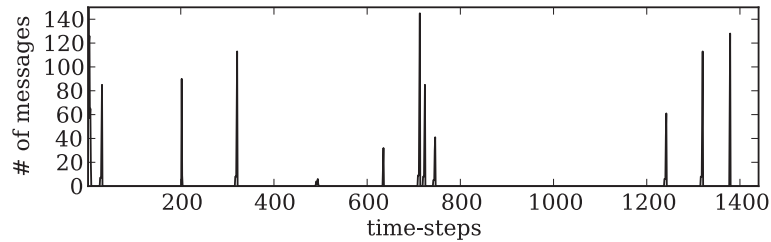
Referring to the agents' instantaneous payoff, it is shown in Figure 15. As it can be observed, the instantaneous payoff in setting 3 is more stable than in setting 2 over time, but is less stable than in setting 1. The reason for this behaviour is that the stability of the payoff is more sensitive to changes in the environment than to the number of agents itself. In the case of setting 3, the average payoff received by the agents throughout the whole simulation was approximately 3.1, which corresponds to an average profit of R\$42.7. Considering that the profit obtained by isolated agents was of R\$39.6, agents in coalitions have obtained a profit 7.8% greater on average. It is important to note that, despite this value being lower than in the other two settings, the average value over time is barely affected during the simulation. In other words, the agents are able to obtain a good payoff even with constant changes in the environment.



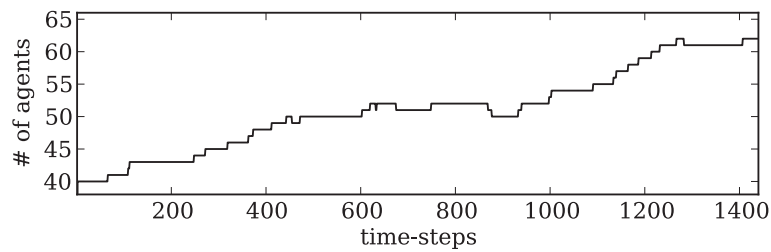
**Figure 9** Variation in the agents' payoff along time, for setting 1. Average instantaneous (left vertical axis) and accumulated (right vertical axis) payoff of agents are shown.



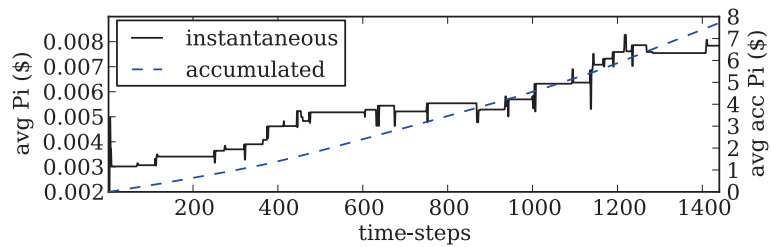
**Figure 10** Average instantaneous duration of coalitions along time, for setting 1.



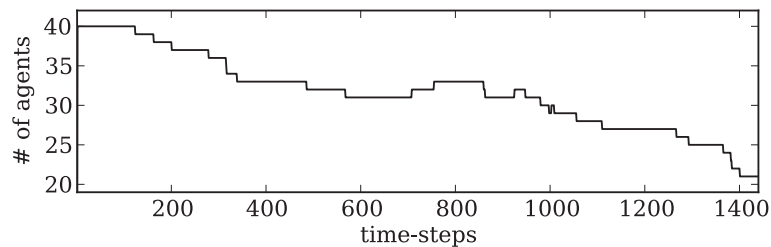
**Figure 11** Amount of messages exchanged among the agents over time, for setting 1.



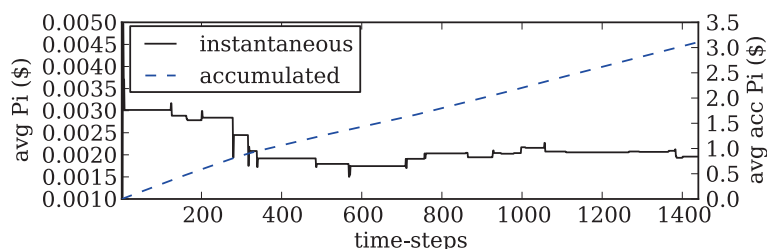
**Figure 12** Number of agents along time, for setting 2.



**Figure 13** Variation in the agents' payoff along time, for setting 2. Average instantaneous (left vertical axis) and accumulated (right vertical axis) payoff of agents are shown.



**Figure 14** Number of agents along time, for setting 3.



**Figure 15** Variation in the agents' payoff along time, for setting 3. Average instantaneous (left vertical axis) and accumulated (right vertical axis) payoff of agents are shown.

## Conclusions

In this paper we have presented DCCF, a distributed method for constrained CSG in smart grids. The proposed approach works by allowing agents to negotiate the formation of coalitions among themselves. On this basis, the agents can propose coalitions to their neighbours or be invited to join a coalition proposed by them.

Based on these experiments, we showed that DCCF is really effective in providing good solutions in a fast, distributed way. We showed that compared to state-of-the-art algorithms, our approach outperforms them in terms of runtime performance by several orders of magnitude, providing solutions whose quality was on average 98.1% of the optimum in tested cases (with standard deviation of 1.4%). In dynamic environments, DCCF also showed good results, being able to keep the agents' payoff sufficiently stable over time. Also, through experiments we showed that compared to isolated agents, the agents who joined coalitions have obtained a higher profit. However, it is important to note that although the results are promising, when compared to other approaches, DCCF is not as generic as them. IP and IDP, for instance, return optimal solutions. Additionally, with an increased number of neighbours (more dense graphs), the agents can take longer to find good results.

For future work, we would like to investigate how the distance constraint could be replaced by a more robust one. In this work, the agent constraints are governed by the physical distance among them. However, more robust alternatives could be used. For example, in order to group agents by the compatibility of their energy profiles, a similarity function could be used. Also, the model could be extended to include more than one kind of constraint.

Another promising future direction would be to extend the model to handle agent uncertainties, e.g., regarding availability. In such scenarios, penalties could be imposed on coalitions that do not fulfil their commitment with the grid. Machine learning techniques could be used to allow the agents to predict their owners' behaviour.

Additionally, reputation mechanisms could be used to rank the agents based on their predictions.

Finally, some changes could be made in the modelling. First, having not just one constraint, but a few, would be a more realistic and interesting approach. Second, agents that are in coalitions could be always checking whether a better coalition is available or not. Probabilistic models could also be incorporated to this end. Finally, other kinds of DERs could be incorporated into the model, allowing the formation of heterogeneous coalitions.

## Endnotes

<sup>a</sup> The distance metric does not play an important role in this work. Anyway, geographical distance is a reasonable approximation (in the absence of a better one) for this problem. In real situations, it can be trivially replaced by another one.

<sup>b</sup> In this work, each time step corresponds to 1 min.

<sup>c</sup> By definition, a power rating of 1 kW over 1 h produces 1 kWh of energy.

<sup>d</sup> In real situations, the ID could be easily replaced by any other comparable code, such as the vehicle's license plate.

<sup>e</sup> The power rating  $W_C$  is obtained from Equation 2. The power rating  $w_i$  was set to 3.3 kW for all agents  $i \in A$ . Such a value was adopted considering that this is the energy transfer rate of some commercial EVs.

<sup>f</sup> Since DCCF finds near-optimal solutions, a comparison against the optimal ones is a useful metric to measure its performance.

<sup>g</sup> In order to improve the visualization of Figure 11, the vertical axis range was set between 0 and 150. In the very beginning, around 300 messages were exchanged.

## Competing interests

The authors declare that they have no competing interests.

## Authors' contributions

The original idea was suggested by JCB. All authors have contributed to the conceptual aspects of this manuscript. The method's design, experimentations, and manuscript's writing, were done by GR under supervision of JCB and AB. All authors read and approved the final manuscript.

## Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable suggestions and comments. We also thank Anderson Tavares for his helpful comments. GR and AB are partially supported by CNPq, FAPERGS and CTIC grants. JCB is supported by the European Regional Development Fund (ERDF), and the Galician Regional Government under projects CN 2012/260 (Consolidation of Research Units: AtlantTIC) and CN 2011/021.

## Author details

<sup>1</sup>Instituto de Informática, Universidade Federal do Rio Grande do Sul, Caixa, Postal 15064, Porto Alegre, Rio Grande do Sul, Brazil. <sup>2</sup>Telematics Engineering Department, University of Vigo, 36310 - Vigo, Spain.

Received: 31 March 2013 Accepted: 12 November 2013

Published: 11 March 2014

## References

1. US Department of Energy (2003) Grid 2030: A national vision for electricity's second 100 years. [http://energy.gov/sites/prod/files/oeprod/DocumentsandMedia/Electric\\_Vision\\_Document.pdf](http://energy.gov/sites/prod/files/oeprod/DocumentsandMedia/Electric_Vision_Document.pdf), accessed 09 September 2011.
2. Kempton W, Tomić J (2005) Vehicle-to-grid power implementation: from stabilizing the grid to supporting large-scale renewable energy. *J Power Sources* 144(1): 280–294
3. Pudjianto D, Ramsay C, Strbac G (2007) Virtual power plant and system integration of distributed energy resources. *Renewable Power Generation*. IET 1(1): 10–16
4. Chalkiadakis G, Robu V, Kota R, Rogers A, Jennings NR (2011) Cooperatives of distributed energy resources for efficient virtual power plants In: Proceedings of the Tenth International Conference on Autonomous Agents and Multiagent Systems (AAMAS'11), Taipei, 2–6 May 2011, pp 787–794
5. Kamboj S, Kempton W, Decker KS (2011) Deploying power grid-integrated electric vehicles as a multi-agent system In: The Tenth International Conference on Autonomous Agents and Multiagent Systems (AAMAS'11), Taipei, Taiwan, 2–6 May 2011, pp 13–20
6. Mihăilescu RC, Vasirani M, Ossowski S (2011) Dynamic coalition adaptation for efficient agent-based virtual power plants. In: Klügl F, Ossowski S (eds) *Multiagent System Technologies*, Lecture Notes in Computer Science. Springer, Berlin Heidelberg, pp 101–112
7. Vasirani M, Kota R, Cavalcante R, Ossowski S, Jennings N (2013) An agent-based approach to virtual power plants of wind power generators and electric vehicles. *IEEE Trans Smart Grids* 4(3): 1314–1322. doi:10.1109/TSG.2013.2259270
8. Sandholm T, Larson K, Andersson M, Shehory O, Tohmé F (1999) Coalition structure generation with worst case guarantees. *Artif Intell* 111(1–2): 209–238
9. Rahwan T, Ramchurn SD, Dang VD, Jennings NR (2007) Near-optimal anytime coalition structure generation In: Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India 6–12 January 2007, pp 2365–2371
10. Rahwan T, Jennings NR (2008) An improved dynamic programming algorithm for coalition structure generation In: Proceedings of the Seventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS'08), Estoril, 12–16 May 2008, pp 1417–1420
11. Horling B, Lesser V (2004) A survey of multi-agent organizational paradigms. *Knowl Eng Rev* 19(4): 281–316
12. Shehory O, Kraus S (1998) Methods for task allocation via agent coalition formation. *Artif Intell* 101(1–2): 165–200
13. Ramos GO, Bazzan ALC (2012) Reduction of coalition structure's search space based on domain information: an application in smart grids In: 2012 Third Brazilian Workshop on Social Simulation (BWSS), Curitiba, 20–23 October 2012, pp 112–119
14. Yeh DY (1986) A dynamic programming approach to the complete set partitioning problem. *BIT Numerical Math* 26(4): 467–474
15. Voice T, Ramchurn SD, Jennings NR (2012) On coalition formation with sparse synergies In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems. International Foundation for Autonomous Agents and Multiagent Systems, Richland, pp 223–230
16. Chalkiadakis G, Markakis E, Jennings NR (2012) Coalitional stability in structured environments In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems. vol. 2: International Foundation for Autonomous Agents and Multiagent Systems, Valencia, Spain, pp 779–786
17. Ueda S, Iwasaki A, Yokoo M, Silaghi M, Hirayama K, Matsui T (2010) Coalition structure generation based on distributed constraint optimization In: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10). AAAI, Atlanta, USA, pp 197–203
18. Chalkiadakis G, Boutilier C (2012) Sequentially optimal repeated coalition formation under uncertainty. *Autonomous Agents Multi-Agent Syst* 24(3): 441–484
19. Ramchurn SD, Polukarov M, Farinelli A, Jennings N, Trong C (2010) Coalition formation with spatial and temporal constraints In: International Joint Conference on Autonomous Agents and Multi-Agent Systems. AAMAS, Toronto, Canada, 10–14 May 2010, pp 1181–1188
20. Ramchurn S, Vytelingum P, Rogers A, Jennings N (2012) Putting the “smarts” into the smart grid: a grand challenge for artificial intelligence. *Commun ACM* 55(4): 86–97
21. Gerding EH, Robu V, Stein S, Parkes DC, Rogers A, Jennings NR (2011) Online mechanism design for electric vehicle charging In: Proceedings of 10th International Conference on Autonomous Agents and Multiagent Systems, Taipei, May, 2011. IFAAMAS, Taipei, Taiwan, 2–6 May 2011, pp 811–818
22. Vandael S, Boucké N, Holvoet T, De Craemer K, Deconinck G (2011) Decentralized coordination of plug-in hybrid vehicles for imbalance reduction in a smart grid In: The 10th International Conference on Autonomous Agents and Multiagent Systems. vol.2: International Foundation for Autonomous Agents and Multiagent Systems, Taipei, Taiwan, pp 803–810

doi:10.1186/1678-4804-20-8

Cite this article as: Ramos et al.: Dynamic constrained coalition formation among electric vehicles. *Journal of the Brazilian Computer Society* 2014 **20**:8.

Submit your manuscript to a SpringerOpen® journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)