Efficient Coalition Structure Generation via Approximately Equivalent Induced Subgraph Games

Filippo Bistaffa*, Georgios Chalkiadakis†, and Alessandro Farinelli‡
*IIIA-CSIC, 08193 Cerdanyola, Spain
†School of Electrical and Computer Engineering, Technical University of Crete, 73100 Chania, Greece
‡Department of Computer Science, University of Verona, 37134 Verona, Italy
Emails: *filippo.bistaffa@iiia.csic.es, †gehalk@intelligence.tuc.gr, ‡alessandro.farinelli@univr.it

***Abstract*—We show that any *characteristic function game* (CFG) G can be *always* turned into an approximately equivalent game represented using the *induced subgraph game* (ISG) representation. Such a transformation incurs obvious benefits in terms of tractability of computing solution concepts for G. Our transformation approach, namely AE-ISG, is based on the solution of a norm approximation problem. We then propose a novel *coalition structure generation* (CSG) approach for ISGs that is based on graph clustering, which outperforms existing CSG approaches for ISGs by using off-the-shelf optimisation solvers. Finally, we provide theoretical guarantees on the value of the optimal CSG solution of G *wrt* the optimal CSG solution of the *approximately equivalent ISG*. As a consequence, our approach allows one to compute approximate CSG solutions with quality guarantees for *any* CFG. Results on a real-world application domain show that our approach outperforms a domain-specific CSG algorithm, both in terms of quality of the solutions and theoretical quality guarantees.**

***Index Terms*—induced subgraph games (ISGs), coalition structure generation (CSG), ridesharing, social networks, graphs**

I. Introduction

Induced subgraph games (ISGs) constitute a celebrated *characteristic function game* (CFG) representation introduced by Deng and Papadimitriou [1], based on a weighted graph among the agents, in which the value of each coalition is the total weight of all its internal edges (i.e., that of the subgraph induced by the coalition). The ISGs representation is *succinct*: any ISG can be completely represented using a polynomial amount of space. In virtue of such simplicity, ISGs have very appealing computational properties, and thus have been widely studied in the cooperative game theory literature [2]. For instance, the *Shapley value* can be immediately computed. Moreover, when the edges' weights are non-negative, one can efficiently check whether an outcome is in the *core* of the game [1]. In this paper we show that the *coalition structure generation* (CSG) problem on ISGs (i.e., determining the optimal partition of the space of agents, the one that maximises the sum of formed coalitions' values), can be solved very efficiently as a *graph clustering* (GC) problem.

Unfortunately, the applicability of ISGs is limited by their simplicity, as ISGs are not capable to represent every possible CFG [2], which, in turn, have been applied to model very important real-world application domains [3], [4]. Indeed, to the best of our knowledge, no actual-world applications of ISGs exist in the literature, and no research to date has examined the problem of how to *approximate* a generic CFG by an ISG, so as to exploit the properties of the ISG model.

Thus, having the means to (approximately) transform a CFG G into an ISG is an interesting direction to investigate, so as to allow one to enjoy the above-mentioned benefits in terms of tractability of computing solution concepts for G.

Motivated by the above discussion, in this paper we achieve this objective by proposing a method to turn *any* graph-restricted [5] and unrestricted game into an *Approximately Equivalent ISG*, namely AE-ISG. AE-ISG is based on a *norm approximation problem* [6], which works with any norm. We discuss how to control the approximation with both soft and hard constraints, which allow one to tailor the approximation quality so as to pursue different objectives (e.g., reduce the overall difference between the values of the coalition in the ISG and in the original CFG, or focus on particular classes of coalitions). Our model can then be solved by off-the-shelf solvers very efficiently. We highlight the equivalence between the CSG problem on ISGs and the graph clustering problem. This allows us to employ a wealth of optimisation techniques for graph clustering so as to tackle CSG on ISGs very efficiently and outperform approaches specifically designed for CSG on ISGs. To the best of our knowledge, this is the first work to highlight and exploit such equivalence. Finally, we provide a technique to compute a theoretical upper bound on the quality of the optimal CSG solution of the original game *wrt* the CSG solution of the ISG, i.e., a method to formally guarantee that the original CSG solution is not better than a given amount *wrt* the one efficiently computed for the ISG. Thus, we provide an efficient method to compute approximate CSG solutions with quality guarantees for *any* CFG.

We test our approach in a real-world domain, i.e., ridesharing. Results show that our approach can transform CFGs with hundreds of agents, while achieving a good approximation quality. Moreover, we provide a highly-parallel GPU version that supports up to 5400 agents (i.e., considering the entire ridesharing dataset, as opposed to 400 agents with the CPU version). Finally, we show that our CSG approach outperforms a recent method proposed by Bistaffa *et al.* [7] that is specifically designed to provide approximate CSG solutions (with theoretical quality guarantees) in the social ridesharing domain. Notice that, in contrast to [7], our approach does not exploit any specific knowledge about the domain.

In Figure 1 we provide an example in which we show how

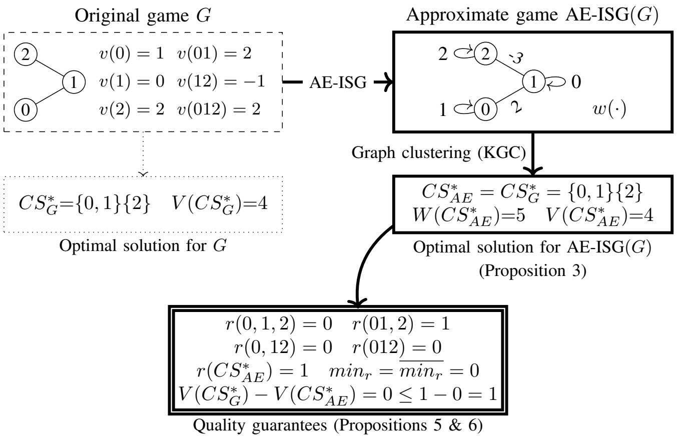
Original game G
2
1
0
$v(0)=1$ $v(01)=2$
$v(1)=0$ $v(12)=-1$
$v(2)=2$ $v(012)=2$
AE-ISG
Approximate game AE-ISG(G)
2
2
-3
1
0
1
0
2
$w(\cdot)$
Graph clustering (KGC)
CS^*_G={0,1}{2} $V(CS^*_G)$=4
Optimal solution for G
$CS^*_{AE}=CS^*_G=\{0,1\}\{2\}$
$W(CS^*_{AE})$=5 $V(CS^*_{AE})$=4
Optimal solution for AE-ISG(G)
(Proposition 3)
$r(0,1,2)=0$ $r(01,2)=1$
$r(0,12)=0$ $r(012)=0$
$r(CS^*_{AE})=1$ $min_r=\overline{min_r}=0$
$V(CS^*_G)-V(CS^*_{AE})=0\leq 1-0=1$
Quality guarantees (Propositions 5 & 6)

Fig. 1. Overview example. The flow of our approach is highlighted in bold.

we compute an (approximate) solution to the CSG problem of a game G. We start by transforming the original game (i.e., the dashed box in the figure) into an AE-ISG one, and then, given that CSG on ISGs is a graph clustering problem, we employ ILP to obtain an optimal solution of the new game. We guarantee that an upper bound on the optimal CSG solution of G (the dotted box in the figure, not computed by our approach) exists, *wrt* to the AE-ISG's computed optimal solution. We provide a summary of the adopted notation in Table I. The technical details will be discussed in the forthcoming sections.

In more detail, this paper makes the following contributions:

- We propose AE-ISG, our approach to turn *any* CFG into an approximately equivalent ISG.
- We provide the computational complexity of AE-ISG and we show that it has *polynomial* complexity when the cardinality of coalitions is bounded by a constant.
- We propose a novel, practically efficient solution algorithm for CSG on ISGs based on graph clustering, and which outperforms two notable CSG approaches [7], [8].
- We formally characterise a bound on the value of the optimal CSG solution of the original CFG *wrt* the CSG solution of the approximated ISG. Thus, our approach allows one to efficiently compute approximate solutions with quality guarantees for *any* CFG.
- We validate our CSG approach on a real-world problem showing that it outperforms a recent method proposed by Bistaffa *et al.* [7], which is specifically designed to provide approximate CSG solutions (with theoretical quality guarantees) in the social ridesharing domain.

Our paper is organised as follows: Section II provides the necessary theoretical background and a review on the related literature with the associated research gap. Sections III and IV discuss the formalisation and the solution of the AE-ISG model. Section V discusses KGC, i.e., our approach to solve CSG on ISGs, and the associated theoretical quality guarantees. Section VI presents our experimental evaluation. Section VII concludes the paper and outlines future work.

Symbol	Definition
G	Characteristic function game
S	Coalition
CS	Coalition structure
$v(S)$	Value of S in G
$V(CS)$	Value of CS in G
CS^*_G	Optimal CSG solution of G
$w_{i,j}$	Weight of edge (i,j)
$w(S)$	Value of S in the ISG
$W(CS)$	Value of CS in the ISG
AE-ISG(G)	Approximately equivalent ISG of G
CS^*_{AE}	Optimal CSG solution of AE-ISG(G)
r_S	Residual associated with S
$r(CS)$	Sum of residuals associated to $S \in CS$
min_r	Minimum $r(\cdot)$ over all coalition structures
$\overline{min_r}$	Lower bound on min_r

TABLE I
NOTATION ADOPTED THROUGHOUT THE PAPER.

II. Background & related work

A. Characteristic function games

A *characteristic function game* G [2] is given by a pair (N, v) where $N = \{1, \ldots, n\}$ is a finite, non-empty set of agents and $v : 2^N \to \mathbb{R}$ is a *characteristic function*, which maps each coalition $S \subseteq N$ to a real number $v(S)$, referred to as the *value* of S. In this paper we focus on *graph-restricted games*, a particular class of characteristic function games first introduced by Myerson [5], and widely studied in the literature [7], [9]. A graph-restricted game is given by a pair $(\mathcal{G}, v)$, where $\mathcal{G} = (N, E)$ is an undirected, connected graph, with $|N| = n$ and $|E| = e$. Myerson considers a coalition S to be feasible if all of their members are connected in the subgraph of $\mathcal{G}$ induced by S. That is, for each pair of players from $i, j \in S$ there is a path in $\mathcal{G}$ that connects them without going out of S. Given $\mathcal{G}$, the set of feasible coalitions is $\mathcal{FC}(\mathcal{G}) = \{S \subseteq N \mid$ The subgraph induced by S on $\mathcal{G}$ is connected$\}$. Notice that every unrestricted CFG can be represented as a graph-restricted game with a fully connected graph. For this reason, in this paper we only refer to graph-restricted games. In addition to such constraints, several application domains inherently enforce a maximum cardinality constraint on the size of coalitions [7], [10], i.e., feasible coalitions cannot have more than k members. Notice that a scenario without cardinality constrains can be seen as a cardinality-constrained one with $k = n$.[1] Thus, in this paper we only refer to cardinality-constrained domains.

B. Induced subgraph games

Induced subgraph games are a simple CFG representation introduced by Deng and Papadimitriou [1]. An ISG is described by an undirected, connected, weighted graph $\mathcal{G}_w = (N, E)$, with $|N| = n$ and $|E| = e$. The weight of edge (i, j) is denoted by $w_{i,j}$. We assume $w_{i,j} \in \mathbb{R}\ \forall (i, j) \in E$.

[1] If $k = n$, the number of coalitions is exponential, hence our transformation approach is no longer polynomial, since all values must be read at least once.

In the ISG that corresponds to $\mathcal{G}_w$, the set of players is N, and the value of a coalition $S \subseteq N$ is defined to be the total weight of all its internal edges, or, in other words, the *subgraph induced by* S, i.e., $w(S) = \sum_{\{i,j\} \subseteq S} w_{i,j}$ such that $(i, j) \in E$. Self-loops are allowed, so that $w(\{i\}) = w_{i,i}$. The value of a coalition structure CS is assessed as the sum of the weights of all the edges induced by CS, i.e., $W(CS) = \sum_{S \in CS} w(S)$.

The representation is *succinct*: any ISG can be completely represented using a polynomial amount of space. In virtue of their simplicity, ISGs have very appealing computational properties. For instance, the *Shapley value* ϕ_i of agent i can be immediately computed as $\phi_i = \frac{1}{2} \sum_{(i,j) \in E} w_{i,j}$. Moreover, when the edges' weights are non-negative, one can efficiently check whether an outcome is in the *core* of the game [1]. ISGs are not capable to represent every possible CFG: in any ISG the value of each coalition is completely determined by the values of its sub-coalitions of size one and two, and in general this is not the case for any CFG [2]. Thus, having the means to (approximately) transform a CFG into an ISG is an interesting direction to investigate.

In terms of coalitional value definition, ISGs can be interpreted as a specific version of the MC-nets representation [11], [12], where a game is described by a collection of rules, and each rule r is of the form $\mathcal{B}_r \rightarrow v_r$, where $\mathcal{B}_r$ is a Boolean formula over a set of variables $\{b_1, \ldots, b_n\}$ and v_r is a real value. The value of a given coalition S is then assessed as the sum of the values of all the rules *applicable* to S, i.e., every rule r such that $\mathcal{B}_r$ is satisfied by the following truth assignment: $b_i =$ **true** if $i \in S$ and $b_i =$ **false** if $i \notin S$. Along these lines, every ISG can be encoded by an MC-net in which, for each edge connecting agents i and j and having a weight $w_{i,j}$, one defines a rule $i \wedge j \rightarrow w_{i,j}$ [12]. However, since ISGs are defined over a graph, graph-based constraints [5] can naturally be applied (as we do in this paper).

One of the fundamental optimisation problems in the context of CFGs is coalition structure generation, discussed below.

C. Coalition structure generation

In its standard definition, the *coalition structure generation* problem [13] takes as input a CFG $G = (N, v)$. Here we assume that CSG takes as input a graph-restricted game $G = (\mathcal{G}, v)$, since we focus on this game representation. A *coalition structure* CS is a partition of N into feasible disjoint coalitions. The set of feasible coalition structures is $\mathcal{CS}(\mathcal{G})$.

The value of a coalition structure CS is assessed as the sum of the values of its composing coalitions, i.e., $V(CS) = \sum_{S \in CS} v(S)$. CSG aims at identifying $CS^* = \arg\max_{CS \in \mathcal{CS}(\mathcal{G})} V(CS)$. Naturally, CSG can also be considered over ISGs also, with the objective of computing the coalition structure CS that maximises $W(CS) = \sum_{S \in CS} w(S)$.

Bachrach *et al.* [14] provide an analysis of the computational aspects of this problem, showing that it is NP-complete. More recently, Leon *et al.* [15] showed that CSG on ISGs can also be solved using *quantum annealing* by casting it as a *quadratic unconstrained binary optimisation* (QUBO) problem. The approach proposed by Leon *et al.* suggests that CSG on ISGs could be solved more efficiently in the future with the development of quantum computing technology. Regardless of advances in quantum computing, in this paper we show that CSG on ISGs can also be tackled very efficiently as a graph clustering problem, discussed hereafter.

D. Graph clustering & graph partitioning

The problem of splitting a given graph into groups of vertices (with many edges joining vertices of the same cluster, and comparatively few edges joining vertices of different clusters) is a fundamental task arising in systems characterised by a network structure [16]. This problem has been widely studied in the literature as *graph clustering* or *graph partitioning* [16]. Despite (or perhaps due to) the widespread presence of this problem in different fields, graph clustering and graph partitioning do not have unique definitions [16].

In this paper we denote the problem of performing CSG on ISGs as a graph clustering one, since they share several features (e.g., the number of groups is not known in advance [17], [18], the objective function is equivalent—see Section V). Throughout this paper, we will adopt the graph clustering formalism provided by Brandes and Erlebach [17], who focus on clustering over weighted graphs. Following the original definition of ISG, we allow edges with negative weights, which express the dissimilarity of the incident nodes in the context of graph clustering [17]. Our graph clustering formalisation of CSG on ISGs will be detailed in Section V.

E. Related work

We now provide a review of the literature related to our work, in the fields of cooperative game theory, graph clustering, mathematical optimisation, and machine learning.

To the best of our knowledge, there is no previous attempt in the *cooperative game theory* (CGT) literature to approximately represent any CFG as an ISG, even though this task has been studied from an *operational research* (OR) perspective as the approximation of a combinatorial function (i.e., the original game's characteristic function) by means of a polynomial function of degree 2 [19]. Certainly, no previous work has investigated the benefits of this transformation to tackle computationally-hard CGT problems (e.g., CSG) in real-world scenarios, as we do in this paper.

Our work is also motivated by the successful application of approaches that approximate a complex input function by means of a more computationally manageable one, achieving a significant performance improvement [20].

There has recently been much work on computing optimal coalition structures or cooperative solution concepts on graph-restricted settings [7], [8], [21]. Now, a strand of literature pioneered by Myerson [5], and later developed by Voice *et al.* [8] and Bistaffa *et al.* [7], [22], has over time proposed CSG algorithms with good properties both in terms of computational complexity and practical applicability. On the one hand, Voice *et al.* [8] proposed an optimal CSG approach for graph-restricted games that satisfy the *independence of disconnected members* (IDM) property (such as ISGs), whose complexity is bounded by the width of the tree decomposition of the graph. On the other hand, Bistaffa *et al.* proposed a CSG approach based on *edge contraction* that has been shown to provide good approximate solutions with theoretical quality guarantees in real-world scenarios such as *collective energy purchasing* [22] and *ridesharing* [7]. Hence, we compare against these two approaches in Section VI.

Furthermore, the work by Bachrach *et al.* [14] provides an in-depth computational complexity analysis of the CSG problem on ISGs (denoted as *weighted graph games*), showing that it is NP-complete for general graphs. Nonetheless, the authors do not discuss any algorithmic approach to the solution of such a problem. Chalkiadakis *et al.* [21] also discussed stability issues arising in graph-restricted games, focusing on the computational complexity of *the core* [2] as a solution concept (considering several graph topologies), without assuming that the grand coalition necessarily forms. To the best of our knowledge, none of the works in the literature has explicitly highlighted the natural link between the CSG problem on ISGs and the graph clustering problem.

Graph clustering is a very mature field offering several solution approaches [16], [23], [24]. However, the computation of exact solutions or approximate solutions with quality guarantees (i.e., the aim of this paper) is not usually pursued. Rather, the focus is on heuristic solutions so as to meaningfully identify clusters in large graphs (e.g., social networks) [16]. For this reason, the standard exact approach to solve graph clustering is based on *integer linear programming* (ILP) [17].

In the field of mathematical optimisation, a wealth of solution techniques have been developed in the last few decades, especially in the context of *convex optimisation*, which comprehends problems such as least-squares and linear programming. It is well known that these problems have a fairly complete theory, arise in a variety of applications, and can be solved numerically very efficiently. The reader can refer to Boyd and Vandenberghe [6] for a comprehensive and in-depth dissertation on this subject.

Learning approaches have also been used in cooperative games. In particular, Balcan *et al.* [25] explore a *probably approximately correct* learning model in cooperative games that aims at learning an *unknown* characteristic function on the basis of several known coalitional values (i.e., samples). Instead, our work aims at approximating (i.e., minimising an error function *over*) the set of coalitional values of a *fully known* function (see Equation 2).

III. The AE-ISG model

Given a graph-restricted CFG G, we aim at computing the ISG that *best* approximates G. More formally, given a graph-restricted game $G = (\mathcal{G}, v)$, the set of feasible coalitions $\mathcal{FC}(\mathcal{G})$, and a norm $\|\cdot\|$, our objective is to define an ISG such that the differences between the values of the coalitions in $\mathcal{FC}(\mathcal{G})$ in the original game and the values in the ISG are minimised, according to $\|\cdot\|$.

To achieve this objective, we consider an ISG based on a weighted graph $\mathcal{G}_w$ with the same vertices and edges of $\mathcal{G}$,[2] and the vector $r \in \mathbb{R}^{|\mathcal{FC}(\mathcal{G})|}$ defined as

$$r_S = \overbrace{\sum_{\substack{(i,j)\in E\\ \{i,j\}\subseteq S}} \{w_{i,j}\}}^{w(S)} - v(S), \quad \forall S \in \mathcal{FC}(\mathcal{G}). \tag{1}$$

Each r_S represents the *difference* between the value of S in the ISG, i.e., $w(S)$, and its value in G, i.e., $v(S)$.

Definition 1 (approximately equivalent ISG). *The approximately equivalent ISG of G, denoted AE-ISG(G), is the ISG in which the weights $w_{i,j}$ are such that the norm $\|r\|$ is minimised.*

Henceforth, AE-ISG(G) is the ISG that *best* approximates G, in the sense that the differences among the values of the coalitions in $\mathcal{FC}(\mathcal{G})$ are minimised, according to $\|\cdot\|$.[3]

The above defined minimisation problem is called an unconstrained[4] *norm approximation problem* (see Boyd and Vandenberghe [6], Chapter 6), which can be represented in the form

$$\text{minimise } \|Aw - v\|, \tag{2}$$

where $v \in \mathbb{R}^{|\mathcal{FC}(\mathcal{G})|}$ is the vector of the values of the coalitions in $\mathcal{FC}(\mathcal{G})$, $w \in \mathbb{R}^e$ is the vector of weights, and $A \in \{0,1\}^{|\mathcal{FC}(\mathcal{G})|\times e}$ is a matrix with $|\mathcal{FC}(\mathcal{G})|$ rows (i.e., one per coalition in $\mathcal{FC}(\mathcal{G})$)[5] and e columns (i.e., one per weight), such that

$$A_{S,w_{i,j}} = \begin{cases} 1, & \text{if } \{i,j\} \subseteq S, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

The vector $r = Aw - v$ is called the problem's *residual*, and each component r_S is called *the residual associated with* S. It is well known that every norm approximation problem is convex and there is always at least one solution [6]. Thus, and since any CFG G corresponds to a graph-restricted game over a fully-connected graph, we can immediately obtain that:

Proposition 1. *Given a CFG G, an AE-ISG(G) always exists.*

A. ℓ_1 and ℓ_2 norms

In this paper we consider the ℓ_1 and the ℓ_2 norms, since they are the most widely used in the optimisation literature [6]. We remark that our approach can be directly used with *all norms*.

Under ℓ_1, Equation 2 represents a *sum of (absolute) residuals approximation problem* [6]. In this case, Equation 2 can be rewritten as

$$\text{minimise} \sum_{S\in\mathcal{FC}(\mathcal{G})} \left| \sum_{\substack{(i,j)\in E\\ \{i,j\}\subseteq S}} \{w_{i,j}\} - v(S) \right|, \tag{4}$$

[2] More precisely, $\mathcal{G}_w$ also includes one self-loop for each node of $\mathcal{G}$, necessary to represent the values of singletons (see Section II-B).

[3] Notice that AE-ISG(G) also represents a quadratic approximation (i.e., an approximation by means of a polynomial function of degree 2) of G's characteristic function [19].

[4] In an unconstrained norm approximation problem each r_S can be arbitrarily large. We investigate the effects of constraining such quantities in Section III-B.

[5] We remark that, since the process of building our AE-ISG model involves one independent operation per row of A, it can be parallelised using up to $|\mathcal{FC}(\mathcal{G})|$ threads.

The weights $w_{i,j}$ that solve Equation 4 can be computed by solving the Linear Programming (LP) problem [6] defined by

$$\begin{aligned} \text{minimise} \quad & \sum_{S \in \mathcal{FC}(\mathcal{G})} t_S, \\ \text{subject to} \quad & \left. \begin{aligned} & \sum_{\substack{(i,j) \in E \\ \{i,j\} \subseteq S}} \{w_{i,j}\} - t_S \leq v(S), \\ & \sum_{\substack{(i,j) \in E \\ \{i,j\} \subseteq S}} \{w_{i,j}\} + t_S \geq v(S), \end{aligned} \right\} \quad \forall S \in \mathcal{FC}(\mathcal{G}). \end{aligned} \tag{5}$$

In the case of ℓ_2, we adopt a similar approach and we rewrite Equation 2 as

$$\text{minimise} \quad \sum_{S \in \mathcal{FC}(\mathcal{G})} \left(\sum_{\substack{(i,j) \in E \\ \{i,j\} \subseteq S}} \{w_{i,j}\} - v(S) \right)^2 . \tag{6}$$

The obtained problem is a *least-squares approximation problem* [6]. This can be solved analytically, and its unique solution is $w = \left(A^T A\right)^{-1} A^T v$.

B. Controlling residuals

In our approximation model with ℓ_1 and ℓ_2 norms, each residual r_S receives the same "emphasis". On the one hand, this approach leads to the best approximation over all coalitions, since the objective of the minimisation (i.e., $\|r\|$) considers all residuals with the same weight. On the other hand, in some cases achieving the best overall approximation might *not* be very meaningful, especially when solving problems such as CSG on the approximated ISG.

In fact, one might want to better approximate a specific class of coalitions, rather than all of them. As an example, in some scenarios CSG solutions are likely to contain singletons, due to the structural properties of the domain.[6] In such domains it is useful to explicitly define the values of singletons in the approximated ISG, since a coalition structure that contains several singletons is a potential optimal CSG solution. This results in CSG solutions of the approximated ISG that better represent the CSG solution of the original game *wrt* the unconstrained approach.

1) Hard constraints: Along these lines, one approach to control the residuals on some coalitions can be achieved by augmenting Equation 2 with a set of (linear) inequalities, obtaining the *constrained* norm approximation problem

$$\begin{aligned} \text{minimise} \quad & \|Aw - v\| \\ \text{subject to} \quad & Aw - v \leq h, \end{aligned} \tag{7}$$

where $h \in \mathbb{R}^{|\mathcal{FC}(\mathcal{G})|}$ specifies the maximum residuals we aim to achieve. It is well known that every norm is a convex function [6], hence Equation 7 is still a *convex* optimisation problem. There is in general no analytical formula for the solution of convex optimisation problems, but (as with LP problems) there are very effective methods for solving them, such as *interior-point* methods [6].

A crucial aspect of the introduction of constraints that bound the residuals is that, in contrast to the unconstrained problem, it is no longer guaranteed that at least one optimal solution exists. This is particularly clear if we consider constraints that enforce *all* residuals to be exactly zero, i.e., force the values of all coalitions in G to be perfectly represented in AE-ISG(G). It is not always possible to perfectly represent a CFG as an ISG (see Section II-B), hence satisfying such constraints is not always possible. Nonetheless, in some scenarios it is possible (and useful) to force an exact representation of a subset of coalitions $\mathcal{R} \subseteq \mathcal{FC}(\mathcal{G})$ that are more interesting than others. We represent this constraint by defining h as

$$h_S = \begin{cases} 0, & \text{if } S \in \mathcal{R}, \\ \infty, & \text{otherwise.} \end{cases}$$

In this case, we can rewrite Equation 7 as

$$\begin{aligned} \text{minimise} \quad & \|Aw - v\| \\ \text{subject to} \quad & A'w = v', \end{aligned} \tag{8}$$

where $v' \in \mathbb{R}^{|\mathcal{R}|}$ is the vector of the values of the coalitions in $\mathcal{R}$, and $A' \in \{0,1\}^{|\mathcal{R}| \times e}$ is defined similarly to Equation 3.

Remark 1. *The constrained norm approximation problem in Equation 8 is solvable iff the system $A'w = v'$ is solvable, i.e., iff the rank of A' is equal to the rank of the augmented matrix $A'|v'$, by applying the Rouché-Capelli theorem.*

Notice that, in case of hard constraints, the values of all the coalitions in $\mathcal{R}$ are fixed (i.e., no longer subject to optimisation). As a consequence, the number of variables in such constrained problem is reduced *wrt* the unconstrained problem, with an obvious improvement in terms of solution runtime. If $\mathcal{R}$ contains only the singletons, $rank(A') = rank(A'|v') = n$, hence it is always possible to compute the optimal solution.

If we enforce constraints on the problem, it might not be possible to solve it (see Remark 1). We now discuss a different approach that is *always* guaranteed to lead to a solution.

2) Weighted norm (soft constrains): An alternative approach to control the residuals is using a *W-weighted norm* $\|r\|_W = \|Wr\|$ [6], where $W \in \mathbb{R}^{|\mathcal{FC}(\mathcal{G})| \times |\mathcal{FC}(\mathcal{G})|}$ is called the *weighting matrix*. W is often diagonal, in which case it gives different relative emphasis to different components of the residual vector r, effectively acting as soft constraints.

A W-weighted norm approximation problem is equivalent to a norm approximation problem with $A_W = WA$ and $b_W = Wb$, hence the techniques and properties discussed in Section III are still applicable. Most importantly, it is now *always* possible to compute an optimal solution. Intuitively, in contrast to hard constraints, soft constraints allow one to "emphasise" classes of coalitions without over-constraining the problem, hence preventing the computation of a solution.

Notice that, even when hard constraints are guaranteed not to over-constrain the problem (e.g., when forcing only singletons to be exact), using a weighted norm is preferable in terms of approximation quality. In fact, since the optimisation problem is less constrained *wrt* the hard-constrained case, the solution is guaranteed to be better in terms of overall quality (i.e., the original CFG is better represented), while still

[6] In ridesharing, if a commuter without a car cannot be accommodated in a car, she will use public transportation (i.e., she will be a singleton).

emphasising singletons and, hence, obtaining better results in terms of CSG solutions.

IV. Solving the AE-ISG model

A wealth of very efficient off-the-shelf optimisation solvers (e.g., CPLEX, Gurobi) can be used to solve norm approximation problems based on ℓ_1 and ℓ_2 norms, since the solution of LP and least-squares problems are very mature technologies [6]. In addition, we provide a highly-parallel solver based on the LSQR algorithm [26] for the least-squares problem in Equation 6, similar in style to the *conjugate gradient* method applied to the least-squares problem.

A. Computational complexity

We formally characterise the worst-case complexity of solving the AE-ISG model in Proposition 2. We remark that this complexity analysis represents a worst-case scenario, and that modern solution algorithms are very efficient and require a lower number of operations, especially by exploiting the sparsity of A [6]. Along these lines, we remark that memory utilisation is a crucial aspect especially when using GPUs [27], whose memory is not as abundant as in the host machine.

Proposition 2. *The worst-case complexity of solving the AE-ISG model when using ℓ_1 (resp. ℓ_2) is $\mathcal{O}\Big((e + |\mathcal{FC}(\mathcal{G})|)^2 \cdot |\mathcal{FC}(\mathcal{G})|\Big)$ (resp. $\mathcal{O}\big(e^2 \cdot |\mathcal{FC}(\mathcal{G})|\big)$).*

Proof. The proof is in the supplementary material. □

The number of edges e is usually small compared to $|\mathcal{FC}(\mathcal{G})|$, hence the above complexity is dominated by $|\mathcal{FC}(\mathcal{G})|$. In general, $|\mathcal{FC}(\mathcal{G})| = 2^n$, if $\mathcal{G}$ is fully connected. Nonetheless, this is just a worst-case scenario, which rarely occurs in actual-world domains. When $\mathcal{G}$ is sparse the number of feasible coalitions is significantly lower [5]. As discussed in Section II, several application domains inherently enforce a maximum cardinality constraint on the size of coalitions. Even without such explicit constraints, it is reasonable to assume that very large coalitions will not form, since the coordination or communication costs involved would be impractical, especially in large-scale scenarios.

As a result, the number of feasible coalitions (and, thus, the complexity of solving the AE-ISG model) is $\mathcal{O}\big(n^k\big)$ [10], i.e., *polynomial* in the number of agents.

V. Solving CSG on ISGs: k-constrained Graph Clustering

Solving the CSG problem on ISGs amounts to partition $\mathcal{G}$ so as to maximise the sum of the weights of the edges induced by each coalition. This task is equivalent to a graph clustering problem, in which each coalition corresponds to a cluster and the optimal solution is given by

$$CS^* = \underset{CS \in \mathcal{CS}(\mathcal{G})}{\arg\max} \underbrace{\sum_{S \in CS} \sum_{\substack{(i,j)\in E \\ \{i,j\} \subseteq S}} w_{i,j}}_{\text{graph clustering objective function}}. \tag{9}$$

Equation 9 highlights the objective function of our graph clustering problem. Notice that maximising such a quantity is equivalent to maximise the *coverage* $\gamma(S)$ of each cluster [17], a standard measure adopted as a graph clustering objective function.[7]

Since we are interested in computing the optimal coalition structure, we adopt the state-of-the-art exact approach to solve graph clustering, i.e., ILP (see Section II-E). Specifically, we put forward KGC, *k-constrained Graph Clustering*, an ILP-based approach to solve the CSG problem. Within KGC, we adopt the formulation of graph clustering proposed by Brandes *et al.* [17], [18]. The ILP is given by n^2 decision variables $X_{i,j} \in \{0, 1\}$, one for every pair of nodes $i, j \in N$.[8] The key idea is that these variables can be interpreted as an equivalence relation (over N) and thus form a clustering (or, equivalently, a coalition structure). $X_{i,j} = 1$ represents the fact that agents i and j are in the same coalition. If such variables are arranged as a matrix, the active variables in the i^{th} row represent all agents in the coalitions that also contain agent i.

To ensure consistency, we enforce the following constraints, which guarantee

$$\begin{aligned} &\text{reflexivity} && \forall i \in N : X_{i,i} = 1, \\ &\text{symmetry} && \forall i, j \in N : X_{i,j} = X_{j,i}, \\ &\text{transitivity} && \forall i, j, z \in N : \begin{cases} X_{i,j} + X_{j,z} - 2 \cdot X_{i,z} \leq 1 \\ X_{i,z} + X_{i,j} - 2 \cdot X_{j,z} \leq 1 \\ X_{j,z} + X_{i,z} - 2 \cdot X_{i,j} \leq 1. \end{cases} \end{aligned} \tag{10}$$

If required, cardinality constraints can be enforced by having at most k active variables in each row, i.e.,

$$\forall i \in N : \sum_{j \in N} X_{i,j} \leq k. \tag{11}$$

Finally, to reflect Equation 9 we maximise the sum of the weights of the edges inside each coalition, i.e, our objective function is

$$\sum_{(i,j)\in E} w_{i,j} \cdot X_{i,j}. \tag{12}$$

Equation 12 encodes the characteristic function of the ISG, i.e., $W(\cdot)$ in the example in Figure 1. Hence, it follows that

Proposition 3. *KGC provides the optimal solution to the CSG problem on ISGs since (10), (11), and (12) encode a graph clustering problem.*

The use of an ILP formulation to solve CSG on ISGs provides clear computational advantages *wrt* alternative approaches (i.e., CFSS [7] and the algorithm by Voice *et al.* [8]).

Remark 2. *While ILP is an NP-complete problem [29], off-the-shelf ILP solvers are able to solve ILP very efficiently by using well-known optimisation techniques, as shown by our*

[7] $\gamma(S) \propto v(S)$, since $\gamma(S) = \frac{v(S)}{\sum_{(i,j)\in E} w_{i,j}}$ and $\sum_{(i,j)\in E} w_{i,j}$ is constant across all S.

[8] This ILP can be simplified by leaving $\binom{n}{2}$ variables and $\binom{n}{3}$ constraints [18]. Fan and Pardalos [28] provide a number of different ILP formalisations for graph partitioning, which, in our case, would still result in a quadratic number of variables.

experiments in Section VI. Indeed, the worst-case time complexities of alternative solution approaches are much higher (i.e., $\mathcal{O}(n^n)$ [7] and $\mathcal{O}(n \cdot \omega^\omega)$ [8], where ω is the width of the adopted tree decomposition). Heuristic GC approaches [16], [23], [24] can also be used. However, by employing a heuristic approach we cannot provide any of the quality guarantees in Section V-A, which rely on the optimality of the GC solution.

A. Quality guarantees on CSG solutions

The purpose of this subsection is twofold. First, we discuss how our ILP-based approach discussed above allows one to compute approximate solutions with given quality guarantees for the CSG problem on ISGs. Second, if such an ISG is the result of the approximation of a CFG G by means of AE-ISG, we discuss how we can provide a theoretical upper bound on the quality of the optimal CSG solution of G *wrt* the CSG solution of the ISG. Notice that, by combining these two contributions, we can compute approximate CSG solutions with quality guarantees for *any* CFG.

1) Approximate CSG on ISGs: Modern ILP solvers, among other solution techniques, also adopt branch-and-bound [30]. This enables the computation of anytime approximate solution with quality guarantees. Formally, given an *optimality gap* $O_{CS} \geq 0$ (often expressed as a percentage value), it is possible to stop the ILP solver as soon as the current solution CS guarantees that

$$W(CS^*_{AE}) - W(CS) \leq O_{CS}, \tag{13}$$

where CS^*_{AE} denotes the optimal CSG solution of AE-ISG(G), and $W(\cdot)$ denotes the characteristic function of AE-ISG(G), i.e., $W(CS)$ computes the sum of the weights of all the edges induced by a given CS within AE-ISG(G). It follows that CS^*_{AE} is the coalition structure with the maximal value of $W(\cdot)$.

Intuitively, the solver stops as soon as the value of CS is within O_{CS} from the value of CS^*_{AE}. Alternatively, it is possible to provide a time budget t_{max} to the solver, which will stop the execution after t_{max} returning an optimality gap O_{CS} that corresponds to the solution found.

2) Quality guarantees on CSG within AE-ISG: In Section III-B, we discussed how to incorporate constraints within AE-ISG so as to produce an optimal CSG solution, CS^*_{AE}, of the ISG that best approximates the original game G—with G's optimal CSG denoted CS^*_G. In this section, we discuss how we provide a theoretical upper bound on the value of CS^*_G *wrt* the value of CS^*_{AE}.

Formally, given a residual vector r resulting from the approximation of a game $G = (\mathcal{G}, v)$ into the corresponding AE-ISG(G), we are interested in assessing a quantity $B_r \geq 0$ that guarantees

$$V(CS^*_G) - V(CS^*_{AE}) \leq B_r. \tag{14}$$

Intuitively, B_r provides an upper bound on the approximation error between the CSG solutions of the original and the approximate game. Hence, B_r depends on the residual vector r, as it contains the approximation error for each coalition. For simplicity, let us initially assume that CS^*_{AE} is *known* to be the optimal CSG solution of AE-ISG(G).[9] CS^*_G, on the other hand, is not known.

To derive B_r on the basis of r, we first prove Lemma 1.

Lemma 1. *Given $G = (\mathcal{G}, v)$, its approximate ISG AE-ISG(G), and the corresponding residual vector r, then*

$$V(CS^*_G) - V(CS^*_{AE}) \leq r(CS^*_{AE}) - r(CS^*_G), \\ \text{where } r(CS) = \sum_{S \in CS} r_S. \tag{15}$$

Proof. The proof is in the supplementary material. □

Remark 3. *Lemma 1 also implies that $r(CS^*_{AE}) - r(CS^*_G) \geq 0$, since $V(CS^*_G) - V(CS^*_{AE}) \geq 0$ (given that CS^*_G is the optimal CSG solution for G and $V(\cdot)$ is G's characteristic function). Thus, $r(CS^*_{AE}) \geq r(CS^*_G)$, i.e., the sum of the residuals for CS^*_{AE} is an upper bound on the sum of the residuals for CS^*_G.*

Notice that, since CS^*_G is not known, $r(CS^*_{AE}) - r(CS^*_G)$ is also unknown. Hence, we compute an upper bound on $r(CS^*_{AE}) - r(CS^*_G)$ as $r(CS^*_{AE})$ minus a lower bound on $r(CS^*_G)$. Such a lower bound is assessed as the *minimum* $r(\cdot)$ over all coalition structures, i.e.,

$$min_r = \min_{CS \in \mathcal{CS}(\mathcal{G})} \sum_{S \in CS} r_S. \tag{16}$$

We discuss the computation of min_r in Section V-A3.

Proposition 4. *Given $G = (\mathcal{G}, v)$, its approximate ISG AE-ISG(G), and the corresponding residual vector r, then*

$$V(CS^*_G) - V(CS^*_{AE}) \leq r(CS^*_{AE}) - r(CS^*_G) \leq \\ \underbrace{r(CS^*_{AE}) - min_r}_{B_r}. \tag{17}$$

Proof. By direct application of Lemma 1. □

If CS^*_{AE} is *not* known (i.e., CSG on AE-ISG(G) has been approximately solved with a given optimality gap), Proposition 4 can be used in conjunction with Equation 13 to derive quality guarantees on the suboptimal CSG solution.

Proposition 5. *Given a CFG G, its approximate ISG AE-ISG(G), and a coalition structure CS solution to the CSG problem of AE-ISG(G) with an optimality gap O_{CS}, then*

$$V(CS^*_G) - V(CS) \leq O_{CS} + r(CS) - min_r \tag{18}$$

Proof. The proof is in the supplementary material. □

Proposition 4 provides interesting insights on the approximation error of CSG solutions within AE-ISG. The difference between $V(CS^*_G)$ and $V(CS^*_{AE})$ is bounded by a value that is higher if we have a high *positive* error for the coalitions in CS^*_{AE} and a high *negative* error for the coalitions in CS^*_G. Intuitively, if the value of the true optimal solution CS^*_G is reduced by a high negative error in AE-ISG, and, in contrast, the value of a different coalition structure CS^*_{AE} is increased

[9] See Proposition 5 for the case with a suboptimal solution CS with a known O_{CS}.

by a high positive error, the difference (i.e., the approximation error) can potentially be very high.

Similarly, Proposition 5 shows that if we only know a suboptimal CS with a given optimality gap O_{CS} (i.e., CS^*_{AE} is unknown), we can still apply this rationale by bounding $r(CS^*_{AE})$ as the sum of $r(CS)$ plus O_{CS}. Notice that, if $O_{CS} = 0$, then $CS = CS^*_{AE}$, and, in such a case, Proposition 5 is correctly equivalent to Proposition 4. Given this,

Proposition 6. *AE-ISG allows one to compute anytime approximate CSG solutions with quality guarantees for* any *CFG.*

3) Computing min_r: The techniques discussed above rely on the computation of a lower bound on $r(CS^*_G)$, represented by the quantity min_r. Unfortunately, computing min_r as defined in Equation 16 is equivalent to solving a *minimisation* CSG problem over the CFG $G_r = (N, v_r)$, with $v_r(S) = r_S$ as its characteristic function. Hence, solving Equation 16 to optimality, while perfectly possible, is not reasonable, since one could solve the original CSG problem over G with the same computational effort.

Instead, we compute a *lower bound*[10] on min_r (denoted as $\overline{min_r}$) by iterating over the set of all the *integer partitions* of n with the largest part that is less or equal than k, namely $\mathcal{P}_{n,k}$. Specifically, we adopt the algorithm proposed by Knuth [31], whose complexity is $\mathcal{O}(|\mathcal{P}_{n,k}| + k)$ [31].

Intuitively, each of these integer partitions is representative for all the coalition structures whose coalitions' sizes correspond to the elements of the partition. As an example, $\langle 5, 5, 3, 2\rangle$ is the integer partition of $n = 15$ representative for all the coalition structures with 2 coalitions of size 5, 1 of size 3, and 1 of size 2. It is well known that $|\mathcal{P}_{n,k}|$ as a function of n is bounded by a polynomial of degree $k - 1$ [32]. Thus,

Proposition 7. *Computing $\overline{min_r}$ has a worst-case time complexity of $\mathcal{O}(n^{k-1})$, i.e.,* polynomial *in the number of agents.*

Since we are interested in computing a lower bound over all coalition structures, each integer partition must be associated to a value that is a lower bound on the values of the represented coalition structures. Formally, given an integer partition $P = \langle p_1, \ldots, p_i, \ldots, p_m\rangle$, we define the function $l_r : \mathcal{P}_{n,k} \to \mathbb{R}$ as $l_r(P) = \sum_{p_i \in P} \min_{\substack{S \in \mathcal{S} \\ |S| = p_i}} r_S$. Then, we compute $\overline{min_r} = \min_{P \in \mathcal{P}_{n,k}} l_r(P)$. The search space of this equation can be further reduced if additional constraints are present.[11] Discarding non-valid integer partitions improves the runtime and allows one to compute a $\overline{min_r}$ that is potentially closer to min_r (i.e., with better quality guarantees).

We now refer the reader back to Figure 1, which provides a full example showing how we compute quality guarantees.

VI. Empirical evaluation

The main goals of our empirical analysis are to:

- evaluate the performance of our approach in terms of runtime and approximation quality;
- investigate the impact of different norms and different constraining methods on the above quantities;
- measure the speed-up that can be achieved by using our GPU solver *wrt* the CPU version;
- compare KGC with the approach by Voice *et al.* [8] in terms of runtime when used to solve CSG on ISGs;
- evaluate the quality of approximate solutions computed by our approach.

In our tests we compare the unconstrained model with both the hard and soft constrained ones. As discussed in Section III-B, we focus on constraints that emphasise residuals corresponding to singletons, since such a class is particularly interesting in the context of ridesharing. In the soft constrained case, we consider a diagonal matrix W such that the weight corresponding to the i^{th} row is $W_{i,i} = 100$ if the i^{th} residual corresponds to a singleton, and $W_{i,i} = 1$ otherwise. Such a value produces the best results for $W_{i,i} \in [10, 10^5]$ in terms of approximation quality, as discussed in the next sections.

Each data point reports the average and the standard error of the mean over 100 random instances. The CPU (resp. GPU) version of AE-ISG is implemented in C++ (resp. CUDA), while KGC is implemented in C++.[12] Our tests are run on a machine with a 3.40GHz CPU, 32GB of memory and a GeForce GTX Titan Xp GPU.

A. Coalitional values

Coalitional values are generated considering the model and the experimental methodology by Bistaffa *et al.* [7]. Specifically, we consider the GeoLife dataset [33] (Figure 2), which comprises 17621 GPS trajectories with a total distance of about 1.2 million km. These trajectories are used to derive the pick-up and drop-off points of our commuting agents. Each coalition (car) S represents a shared ride that transports each member from its pick-up to its drop-off point. Thus, the value of S (namely, $v(S)$) represents the total travel cost (measured in €) associated to the length of the shortest route L^*_S travelled by S. More formally,

$$v(S) = \begin{cases} K_{pt}, & \text{if } S \text{ is a rider without a car,} \\ K_{fuel} \cdot L^*_S, & \text{otherwise,} \end{cases} \tag{19}$$

where $K_{fuel} = -0.0\overline{6}$ €/km (considering a fuel cost of -1 € per litre and an average consumption of 1 litre of fuel every 15 km) and $K_{pt} = -3$ €, which represents the cost of a public transportation ticket, e.g., a bus or a train ticket. Following Bistaffa *et al.* [7], we compute L^*_S as follows. We consider the set of pick-up and drop-off points for n agents. We then consider the matrix $D \in \mathbb{R}^{2\cdot n \times 2\cdot n}$, such that $D_{i,j}$ represents the length of the shortest path from point i to point j.

We consider the set $\mathcal{L}_S$ of all sequences of pick-up and drop-off points of the agents of S in which (i) the first (resp. the last) point of $L \in \mathcal{L}_S$ is the pick-up (resp. drop-off) point of the currently designated driver, and (ii) for each request

[10] Sandholm *et al.* [13] also proposed a method to compute a bound on the optimal CSG solution. Their bound is based on the value of the grand coalition and all the coalition structures with at least one coalition of size $n-1$. This approach is not applicable to cardinality-constrained scenarios, in which these coalition structures are not feasible and not associated to a value.

[11] In the ridesharing domain, non-singletons must contain at least one driver.

[12] Our source code is available at https://github.com/filippobistaffa/APEQIS and https://github.com/filippobistaffa/KGC.

Fig. 2. The map of Beijing derived from the GeoLife dataset.

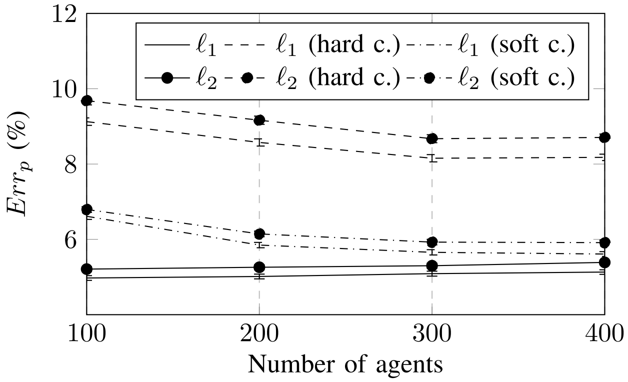

Fig. 4. Percentage error Err_p.

$r \in S$, the pick-up point of r precedes the drop-off point of r in $L \in \mathcal{L}_S$. The optimal route for S, namely L^*_S, is then defined as the one that minimises the sum of the lengths $D_{i,j}$ for every couple (i, j) of subsequent points in L, for every route $L \in \mathcal{L}_S$, and considering all the possible drivers in S.

B. AE-ISG runtime

We evaluate the runtime needed to solve the AE-ISG model, both on the CPU (i.e., with CPLEX) and on the GPU (i.e., with CUDA). We fix an overall runtime limit of 10^4 seconds and we increase the number of agents until such limit is reached.

Figure 3 shows that our (unconstrained) CPU approach scales up to 400 agents within the runtime limit, and that, as expected, the introduction of hard constraints has a positive impact on performance, reducing the runtime by one order of magnitude. On the other hand, the use of the W-weighted norm has no impact on the runtime, since it does not reduce the number of variables *wrt* the unconstrained model. For this reason, we do not show such results. Figure 3 also highlights the significant speed-up produced by GPU parallelism in the case of ℓ_2. Specifically, our GPU version achieves an average speed-up of 260× in the unconstrained model, and 61× in the constrained one. In the best case (unconstrained ℓ_2), such speed-up is 350×. Notice that the speed-up grows when the size of the optimisation problem grows, showing that our GPU implementation successfully exploits the increasing degree of parallelism. Thanks to this, our GPU approach can scale up to problem sizes considering the entire dataset (not depicted here), and therefore it is a viable approach for the transformation of large-scale problems.

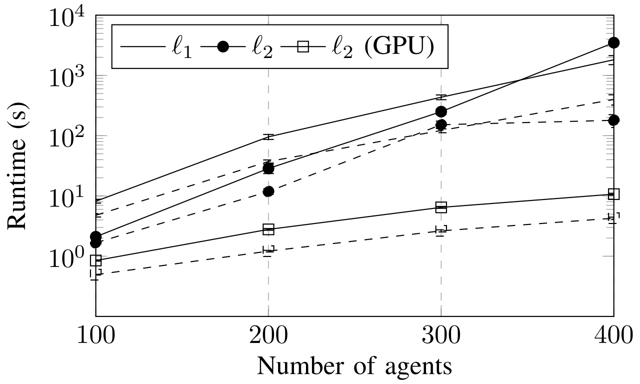

Fig. 3. AE-ISG runtime. Solid (resp. dashed) lines refer to the unconstrained (resp. hard-constrained) approach.

C. AE-ISG quality

The approximation error for each coalition is directly represented by the residual vector r, whose norm $\|r\|$ is the object of the minimisation. On the basis of the optimal r, we can infer natural quality measures such as the *percentage error wrt* the total sum of the coalitional values, i.e., $Err_p{=}100 \cdot \frac{\|r\|_1}{\sum_{S \in \mathcal{S}} v(S)}$ or the *average coalitional error* $Err_a = \frac{\|r\|_1}{|r|}$. Figure 4 shows that, for the unconstrained approach, Err_p is $\sim$5%, corresponding to $Err_a = 0.45$€. As expected, the introduction of constraints produces a degradation in terms of overall approximation quality. When the value of singletons is fixed (i.e., hard-constrained), Err_p is $\sim$9%, corresponding to $Err_a = 0.78$€. On the other hand, the use of W-weighted norm results in Err_p =$\sim$6%, corresponding to $Err_a = 0.53$€, confirming that soft constraints produce a better overall quality. Finally, our results show that ℓ_1 and ℓ_2 are comparable in terms of approximation quality. Therefore, we conclude that, within AE-ISG, ℓ_2 is preferable to ℓ_1, since it provides a benefit in terms of runtime performance thanks to GPU parallelism. As a consequence, we only consider ℓ_2.

D. KGC runtime

We now compare KGC with the approach by Voice *et al.* [8] in terms of runtime when used to optimally solve CSG on ISGs. Our choice is motivated by the fact that, to the best of our knowledge, [8] is the only approach designed to optimally solve CSG on ISGs. In the supplementary material we compare one of the most notable GC approaches, i.e., one based on the concept of *dominant set* [23], with KGC. As expected, the heuristic approach is much faster in terms of runtime since it does not seek optimality. Of course, its performance is sub-optimal: in our tests, its solution quality is 84% *wrt* the optimal solution computed by KGC. Moreover, we remark that by replacing KGC with a heuristic approach one loses all the theoretical quality guarantees we provide in Section V-A.

Tree width (ω)	Average speed-up *wrt* [8]
7	$10.90\times$
8	$201.72\times$
9	$5237.16\times$
10	$145634.53\times$

TABLE II
SPEED-UP OF KGC *wrt* VOICE *et al.* [8].

We consider 100 ISG instances obtained by means of AE-ISG, using the same methodology discussed in Section VI. For this test, we consider $n = 60$, as larger ISG instances could not be solved by [8] in a reasonable amount of time. We then measure the runtime of the two approaches when solving the same ISG instance. Table II reports the average speed-up (i.e., the ratio between the runtime of KGC and the one of [8]), depending on the width ω of the considered tree decomposition (as such parameter determines the computational complexity of [8]). Results show that KGC outperforms the counterpart approach by providing a speed-up as large as $145634.53\times$, in virtue of the fact that its performance is not affected by ω.

E. Approximate CSG solutions with quality guarantees

We now proceed to evaluate our CSG approach in terms of quality of the approximate CSG solution it provides. We consider a real-world problem domain, i.e., social ridesharing [7], and we show that it outperforms the approximate method proposed by Bistaffa *et al.* [7], namely SR-CFSS, which was specifically designed for that particular domain, and which was shown to provide good results. SR-CFSS is a branch-and-bound search-based CSG approach that can be either run to completion (to compute the optimal solution), or stopped after a given *time budget*. In such a case, SR-CFSS returns an approximate solution and a theoretical bound for the value of the optimal one.

To ensure a fair comparison, we measure the runtime required by our approach, and we provide the same time budget to the approximate version of SR-CFSS. For this experiment, we consider $n = 100$. Following Bistaffa *et al.* [7], [22], we consider the *Maximum Performance Ratio* (MPR), a standard measure to quantify the quality of approximate algorithms. Note that here we consider CSG as a *minimisation* problem, since in the ridesharing domain coalitional values represent costs that have to be minimised. Formally, given a game G, we denote the approximate CSG solution as $Approx(G)$ and the lower bound for the value of the optimal CSG solution as $Bound(G)$, i.e., $Bound(G)$ is a value such that the value of the optimal solution cannot be lower (i.e., better). In the case of AE-ISG, $Bound(G) = Approx(G) - B_r$ (cf. Equation 14). Then, in the case of minimisation, the MPR is defined [7] as

$$MPR = \frac{Approx(G)}{Bound(G)}.$$

Figure 5 shows that, both for the hard and soft-constrained case, our approach provides better solutions (i.e., with lower costs). Moreover, when using soft constraints, AE-ISG provides a bound for the optimal that is slightly better than the one provided by SR-CFSS. Given this, in the soft-constrained case the MPR associated to AE-ISG outperforms the one associated to the counterpart approach. Notice that AE-ISG does not possess any specific knowledge about the characteristic function, while SR-CFSS is specifically devised for ridesharing.

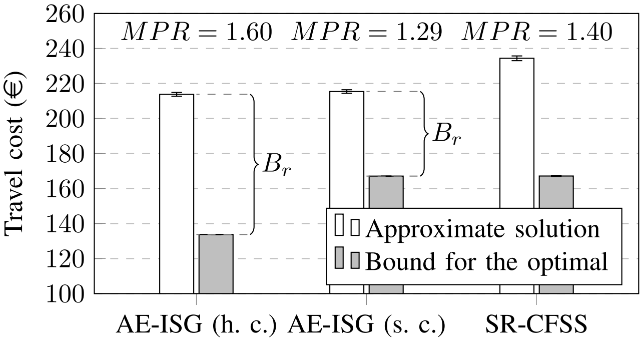

Fig. 5. Approximate solutions with quality guarantees (lower MPR is better).

VII. CONCLUSIONS

We introduced the AE-ISG model, which allows one to turn *any* CFG into an approximately equivalent ISG, in which CSG solutions can be computed by means of our graph clustering approach. We also provided theoretical guarantees on the value of the original CSG solution *wrt* the CSG solution of the approximated ISG, allowing one to compute approximate CSG solutions with quality guarantees for *any* CFG.

In future work, we intend to investigate the impact of a partially unknown characteristic function on the approximation quality. We also aim to apply our approach to other important real-world scenarios—for instance on that of collective energy purchasing in smart grid scenarios [22]. To this end, it would be interesting to combine our approach with explicit bargaining or bidding by the agents interested in forming coalitions.

More generally, our work in this paper can naturally lead to interesting theoretical extensions beyond the coalition structure generation problem. As an example, we note that promising preliminary results show that the computational advantages provided by the AE-ISG paradigm could be fruitfully exploited to compute Owen values [34], by employing a two-step procedure based on Shapley value computation (which can be efficiently computed on ISGs). Along those lines, we aim at further investigating the theoretical properties of Owen values in the context of ISGs; and to extend the computational benefits and quality guarantees provided by our model to other game-theoretic solution concepts, including those used for settings with overlapping coalitions [35], [36].

ACKNOWLEDGMENT

This work was supported by project CI-SUSTAIN funded by the Spanish Ministry of Science and Innovation (PID2019-104156GB-I00). This project has received funding from the EU H2020 programme under grant agreement #769142. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU.

References

[1] X. Deng and C. Papadimitriou, "On the complexity of cooperative solution concepts," *Mathematics of Operations Research*, vol. 19, no. 2, pp. 257–266, 1994.

[2] G. Chalkiadakis, E. Elkind, and M. Wooldridge, *Computational Aspects of Cooperative Game Theory*, ser. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan and Claypool Publishers, 2011.

[3] J. Wu and M. Zhang, "A game theoretical approach to optimal control of dual drug delivery for hiv infection treatment," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 40, no. 3, pp. 694–702, 2010.

[4] F. Seo and M. Sakawa, "A game theoretic approach with risk assessment for international conflict solving," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, no. 1, pp. 141–148, 1990.

[5] R. B. Myerson, "Graphs and cooperation in games," *Mathematics of Operations Research*, vol. 2, no. 3, pp. 225–229, 1977.

[6] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[7] F. Bistaffa, A. Farinelli, G. Chalkiadakis, and S. D. Ramchurn, "A cooperative game-theoretic approach to the social ridesharing problem," *Artificial Intelligence*, vol. 246, pp. 86–117, 2017.

[8] T. Voice, M. Polukarov, and N. R. Jennings, "Coalition structure generation over graphs," *Journal of Artificial Intelligence Research*, vol. 45, pp. 165–196, 2012.

[9] E. Elkind, "Coalitional games on sparse social networks," in *International Conference on Web and Internet Economics*, 2014, pp. 308–321.

[10] O. Shehory and S. Kraus, "Methods for task allocation via agent coalition formation," *Artificial Intelligence*, vol. 101, no. 1, pp. 165–200, 1998.

[11] S. Ieong and Y. Shoham, "Marginal contribution nets: a compact representation scheme for coalitional games," in *ACM Conference on Electronic Commerce*, 2005, pp. 193–202.

[12] T. Rahwan, T. P. Michalak, M. Wooldridge, and N. R. Jennings, "Coalition structure generation: A survey," *Artificial Intelligence*, vol. 229, pp. 139–174, 2015.

[13] T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohmé, "Coalition structure generation with worst case guarantees," *Artificial Intelligence*, vol. 111, no. 1, pp. 209–238, 1999.

[14] Y. Bachrach, P. Kohli, V. Kolmogorov, and M. Zadimoghaddam, "Optimal coalition structure generation in cooperative graph games," in *AAAI Conference on Artificial Intelligence*, 2013, pp. 81–87.

[15] F. Leon, A.-Ş. Lupu, and C. Bădică, "Multiagent coalition structure optimization by quantum annealing," in *Conference on Computational Collective Intelligence Technologies and Applications*, 2017, pp. 331–341.

[16] S. Fortunato, "Community detection in graphs," *Physics reports*, vol. 486, no. 3, pp. 75–174, 2010.

[17] U. Brandes and T. Erlebach, *Network analysis: methodological foundations*. Springer Science & Business Media, 2005, vol. 3418.

[18] U. Brandes, D. Delling, M. Gaertler, R. Gorke, M. Hoefer, Z. Nikoloski, and D. Wagner, "On modularity clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 2, pp. 172–188, 2008.

[19] P. L. Hammer and R. Holzman, "Approximations of pseudo-boolean functions; applications to game theory," *Zeitschrift für Operations Research*, vol. 36, no. 1, pp. 3–21, 1992.

[20] D. Berjón, G. Gallego, C. Cuevas, F. Morán, and N. García, "Optimal piecewise linear function approximation for gpu-based applications," *IEEE Transactions on Cybernetics*, vol. 46, no. 11, pp. 2584–2595, 2016.

[21] G. Chalkiadakis, G. Greco, and E. Markakis, "Characteristic function games with restricted agent interactions: Core-stability and coalition structures," *Artificial Intelligence*, vol. 232, pp. 76–113, 2016.

[22] F. Bistaffa, A. Farinelli, J. Cerquides, J. Rodríguez-Aguilar, and S. D. Ramchurn, "Algorithms for graph-constrained coalition formation in the real world," *ACM Transactions on Intelligent Systems and Technology*, vol. 8, no. 4, 2017.

[23] M. Pavan and M. Pelillo, "Dominant sets and pairwise clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 167–172, 2006.

[24] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[25] M.-F. Balcan, A. D. Procaccia, and Y. Zick, "Learning cooperative games," in *International Joint Conferences on Artificial Intelligence*, 2015, pp. 475–481.

[26] C. Paige and M. Saunders, "LSQR: An algorithm for sparse linear equations and sparse least squares," *ACM Transactions on Mathematical Software*, vol. 8, no. 1, pp. 43–71, 1982.

[27] F. Bistaffa, N. Bombieri, and A. Farinelli, "An efficient approach for accelerating bucket elimination on gpus," *IEEE Transactions on Cybernetics*, vol. 47, no. 11, pp. 3967–3979, 2017.

[28] N. Fan and P. M. Pardalos, "Linear and quadratic programming approaches for the general graph partitioning problem," *Journal of Global Optimization*, vol. 48, no. 1, pp. 57–71, 2010.

[29] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity*. Courier Corporation, 1982.

[30] R. S. Garfinkel and G. L. Nemhauser, *Integer programming*. Wiley New York, 1972.

[31] D. E. Knuth, *The Art of Computer Programming, Volume 4, Fascicle 3: Generating all combinations and partitions*. Addison-Wesley Professional, 2005.

[32] P. Flajolet and R. Sedgewick, *Analytic combinatorics*. Cambridge University Press, 2009.

[33] Microsoft Research. (2009) Geolife dataset. [Online]. Available: http://research.microsoft.com/en-us/projects/geolife

[34] S. Béal, A. Khmelnitskaya, and P. Solal, "Two-step values for games with two-level communication structure," *Journal of Combinatorial Optimization*, vol. 35, no. 2, pp. 563–587, 2018.

[35] G. Chalkiadakis, E. Elkind, E. Markakis, M. Polukarov, and N. R. Jennings, "Cooperative games with overlapping coalitions," *Journal of Artificial Intelligence Research*, vol. 39, pp. 179–216, 2010.

[36] M. Mamakos and G. Chalkiadakis, "Probability bounds for overlapping coalition formation," in *International Joint Conference on Artificial Intelligence*, 2017, pp. 331–337.

Filippo Bistaffa received the Ph.D. in Computer Science from the University of Verona in 2016. He is currently a Post-Doctoral Research Fellow (former Marie Skłodowska-Curie Fellow) at the Artificial Intelligence Research Institute (IIIA-CSIC), Bellaterra, Spain. His research interests comprise combinatorial optimisation problems for realistic applications (such as ridesharing and team formation) and GPU computing.

Georgios Chalkiadakis received his Ph.D. degree in Computer Science from the University of Toronto in 2007. He is an Associate Professor in the School of Electrical and Computer Engineering, Technical University of Crete. Georgios is a co-author of the textbook "Computational Aspects of Cooperative Game Theory" (Morgan & Claypool, 2011). His current research focuses on various aspects of multiagent systems, and more specifically on: sequential decision making under uncertainty, reinforcement learning in multiagent environments, recommender systems, coalition formation, and mechanism design.

Alessandro Farinelli received the Ph.D. degree from the University of Rome La Sapienza in 2005. He is a Full Professor at University of Verona, Department of Computer Science. His research interests focus on developing novel Artificial Intelligence methodologies applied to Robotics and Cyber-Physical Systems. In particular, he focuses on multiagent coordination, decentralised optimisation, reinforcement learning and data analysis for Cyber-Physical Systems.