GCS-Q: Quantum Graph Coalition Structure Generation

Supreeth Mysore Venkatesh[2], Antonio Macaluso[1⋆], and Matthias Klusch[1]

[1] German Research Center for Artificial Intelligence (DFKI), Saarbruecken, Germany
{antonio.macaluso, matthias.klusch}@dfki.de
[2] University of Saarland, Saarbruecken, Germany
s8sumyso@stud.uni-saarland.de

Abstract. The problem of generating an optimal coalition structure for a given coalition game of rational agents is to find a partition that maximizes their social welfare and known to be NP-hard. Though there are algorithmic solutions with high computational complexity available for this combinatorial optimization problem, it is unknown whether quantum-supported solutions may outperform classical algorithms.
In this paper, we propose a novel quantum-supported solution for coalition structure generation in Induced Subgraph Games (ISGs). Our hybrid classical-quantum algorithm, called GCS-Q, iteratively splits a given n-agent graph game into two nonempty subsets in order to obtain a coalition structure with a higher coalition value. The GCS-Q solves the optimal split problem $\mathcal{O}(n)$ times, exploring $\mathcal{O}(2^n)$ partitions at each step. In particular, the optimal split problem is reformulated as a QUBO and executed on a quantum annealer, which is capable of providing the solution in linear time with respect to n. We show that GCS-Q outperforms the currently best classical and quantum solvers for coalition structure generation in ISGs with its runtime in the order of n^2 and an expected approximation ratio of 93% on standard benchmark datasets.

Keywords: Coalition Formation · Quantum Computing · Quantum Artificial Intelligence · Multi-Agent Systems

1 Introduction

One major challenge of rational cooperation in multi-agent systems is to solve the coalition structure generation (CSG) problem. Given a coalition game (A, v) with a set A of n agents and a characteristic function $v : \mathcal{P}(A) \to \mathbb{R}$ for coalition values $v(C)$ for all non-empty coalitions C in A, the problem is to find a coalition structure CS^* of A that maximizes the social welfare $\sum_{C \in CS^*} v(C)$. This combinatorial optimization (partitioning) problem is known to be NP-complete [1] with an exponential number of possible coalition structures. Several solution methods for the CSG problem exist, such as the currently best solver BOSS [5] with run-time complexity of $O(3^n)$, that avoid exploring the complete search

⋆ Corresponding author

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_11

space to find an optimal solution. In the following, we focus on the CSG problem for Induced Subgraph Games. In this case, the coalition game is induced by an undirected weighted graph where the agents are denoted as nodes and the coalition values are the sum of the weights of edges between coalition members in the graph. However, the problem remains NP-complete [2] and therefore intractable for large values of n in practice.

One open question is whether the usage of quantum computational means may contribute to solve this problem faster than it is possible with the currently best state-of-the-art solvers. To this end, we developed a novel quantum-supported solution, called GCS-Q, for solving the coalition structure generation problem for Induced Subgraph Games. In particular, the GCS-Q leverages classical and quantum computation for an approximate, anytime solution of the problem, and is inspired by divisive hierarchical clustering. The GCS-Q starts with the grand coalition and iteratively splits it up until the coalitions are singleton sets such that it builds the hierarchy in $n-1$ steps for a coalition game with n agents and, at each step, explores $\mathcal{O}(2^n)$ partitions. However, the GCS-Q identifies the optimal split for a given coalition using a quadratic unconstrained binary optimization (QUBO) problem formulation, which can be solved experimentally on a real quantum annealing device in linear time. As a result, the overall time complexity of the GCS-Q is in the order of n^2, significantly outperforming the currently approximate classical state-of-the-art solvers. We conduct our comparative performance evaluation of the GCS-Q with selected classical baselines on standard benchmark data using a D-Wave 2000Q quantum annealer and sufficient worst-case approximation ratio.

The remainder of the paper is structured as follows: The problem of coalition structure generation for graph games is formalized in section 2, followed by a discussion of related work in section 3. Our quantum-supported solution GCS-Q is described in section 4 and its performance comparatively evaluated in section 5. Section 6 concludes with a summary of achievements and future work.

2 Problem Formulation

As mentioned above, the coalition structure generation problem is to find a partition or coalition structure of a given set of rational agents whose value or social welfare is maximal for a given coalition game. We focus on the *Induced Subgraph Games (ISGs)* that are based on graph-restricted games, i.e., coalition games induced by connected, undirected weighted graphs (cf. Definition 1) [17].

Definition 1. *An Induced Subgraph Game (A, v) is induced by a connected, undirected, weighted graph $G(V, w)$: The set $V = \{i\}_{i\in\{1..n\}}$ of nodes in G represents the set $A = \{a_1, a_2, ...a_n\}$ of n agents, and the real valued weights $w_{i,j}$ of edges (i, j) in w denote the synergies of cooperation or joint utilities of agents $a_i, a_j \in A$ in feasible coalitions $C \subseteq A$. A coalition C is feasible if and only if it induces a connected subgraph of G. The coalition value $v(C)$ of a feasible*

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_11

coalition C is

$$v(C) = \sum_{(i,j)\in w,\ i,j\in C} w_{i,j}. \tag{1}$$

For given ISG (A, v), coalition structures CS are partitions of A into mutually disjoint, feasible coalitions C. The coalition structure generation problem for a given ISG is to find the optimal coalition structure CS^* with maximal coalition value (or social welfare) for (A, v):

$$CS^* = \arg\max_{CS} \sum_{C\in CS} v(C) \tag{2}$$

In ISGs, the coalition values depend only on the pairwise interactions between agents represented in the graph, but the set of possible solutions is not restricted and the problem remains NP-complete [2]. However, if $G(V, w)$ is a connected graph and the edge weights are all positives, i.e., $(w_{i,j}) \in \mathbb{R}^+$, the grand coalition $g_c = A$ is always the optimal coalition structure [1].

In this paper, we refer to the CSG problem in ISG as the *ISG problem* and we assume a fully connected graph with positive and negative edge weights.

3 Related works

There are two broad classes of solutions for solving a CSG problem. *Exact* methods operate to find the global optimum by exploring a large number of possible coalition structures. Contrarily, *approximate* methods reformulate the original problem to find near-optimal solutions. Any algorithmic solution (exact or approximate) can be characterized based on the minimum time required to provide the solution. Anytime optimal algorithms (e.g., IP [13]) generate an initial set of possible solutions within a bound from the optimal and then improve their quality iteratively. The downside is that these algorithms might end up searching the entire space of all possible coalition structures, which translates into a worst-case time complexity of $\mathcal{O}(n^n)$ for n agents. Nevertheless, such methods allow intermediate solutions during execution, which can be critical for many real-world scenarios. An alternative approach consists of using Dynamic Programming [12], which avoids exploring the entire solution space without losing the guarantees of finding the optimal coalition structure. However, these approaches must be executed entirely to obtain the final solution. Nonetheless, algorithms within this category represent state-of-the-art in terms of worst-case time complexity ($\mathcal{O}(3^n)$) when considering any generic coalition game with no prior knowledge of the nature of the characteristic function [11,5]. The method DyCE (Dynamic programming for optimal connected Coalition structure Evaluation) [18] is based on IDP [12] and exact but not anytime. Its memory requirements are of the order of $\mathcal{O}(2^n)$ with a reported limitation of application to up to 30 agents in practice.

Approximate methods such as C-link (Coalition-link) [6] solve the CSG problem based on agglomerative clustering. C-link considers only the values of coalitions up to size two as interaction scores between the two agents by discarding

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_11

other coalition values in the input. C-Link scales as $\mathcal{O}(n^3)$ with an estimated worst-case approximation ratio of 80% on custom datasets.

In the context of ISGs, different approaches assume specific graph structures to improve the runtime for the final solution. For example, CFSS [4] is an anytime solver for graph games that uses a branch-and-bound search technique in the solution space. CFSS has been used for comparative evaluation with the algorithms mentioned above and has shown excellent results for sparse graphs. However, the worst-case computational complexity of CFSS is still of the order $\mathcal{O}(n^n)$. Lately, the idea of applying *k-constrained Graph Clustering (KGC)* formulated as *Integer Linear Programming (ILP)* has been investigated [3]. This approach allows efficient implementation when dealing with sparse graphs.

Recently, the first general hybrid quantum-classical algorithm for solving any generic coalition game has been formulated. The algorithm, named BILP-Q [16], is suitable for gate-based quantum computing and quantum annealing and shows a complexity of the order of $\mathcal{O}(2^n)$ in the case of constrained CSG. However, BILP-Q requires the number of logical qubits to be exponential in n, which is a significant limitation considering near-term quantum technology.

4 Methods

In this section, we present GCS-Q (*Quantum-supported solution for Graph-restricted Coalition Structure generation*), a novel quantum-supported anytime approximate algorithm for CSG in the context of ISGs. GCS-Q starts from the grand coalition as the initial coalition structure and recursively performs a split to find the best bipartition of the agents based on the graph induced by the coalition game. This approach's primary source of complexity is given by finding the optimal split at each step, which is NP-hard. However, properly formulating this problem and delegating it to a quantum annealer allows for obtaining a quantum-supported algorithm that runs quadratically in the number of agents and can outperform state-of-the-art classical and quantum solutions.

4.1 Optimal Split

Given a coalition game (A, v), where $A = \{a_1, a_2, ...a_n\}$ is a set of agents of size $|A| = n$ and $v : \mathcal{P}(A) \to \mathbb{R}$ is a characteristic function, a split $\{C, \overline{C}\}$ is defined as the bipartition of A into two disjoint subsets $C, \overline{C}$.

Definition 2. *Finding the optimal split of a coalition game (A, v) is an optimization problem of the form:*

$$\arg\max_{C,\overline{C}} \; v(\{C, \overline{C}\}) = v(C) + v(\overline{C}) \tag{3}$$

$$s.\ t. \quad C \cup \overline{C} = A \quad and \quad C \cap \overline{C} = \emptyset. \tag{4}$$

The exhaustive enumeration of all possible splits for a coalition game of n agents is of the order $\mathcal{O}(2^n)$.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_11

For a given ISG, the optimal split into two mutually disjoint sets translates into the minimum cut problem of the graph underlying the coalition game.

Definition 3. *Let be $G(A, w)$ a weighted undirected graph. A cut is a partition of the vertices into two sets C and $\overline{C}$ such that $\overline{C} = A - C$. The value of a cut $\delta(S)$, where $S = \{C, \overline{C}\}$, is defined as the sum of the edge weights connecting the nodes of the two sets C and $\overline{C}$:*

$$\delta(C, \overline{C}) = \sum_{i \in C, j \in \overline{C}} w_{i,j}. \tag{5}$$

The weighted minimum cut (min-cut) is an optimization problem that aims to find a cut with minimum value:

$$\textit{min-cut}\ (G) = \arg\min_{S} \delta(S). \tag{6}$$

Thus, we establish the equivalence between the optimal split (cf. Definition 2) and *min-cut* (cf. Definition 3) in the context of ISGs.

Lemma 1. *Given an ISG (A, v) with an underlying graph $G(A, w)$, finding the optimal split for (A, v) is equivalent to solving the min-cut for $G(A, w)$.*

Proof. The value of a coalition A for a given coalition game can be considered constant and calculated as the sum of all the edge weights between the agents into the coalition:

$$v(A) = \sum_{i,j \in A} w_{i,j}. \tag{7}$$

A cut $\delta(S)$ on $G(A, w)$, where $S = \{C, \overline{C}\}$, produces two independent sets of nodes representing two separate coalitions (cf. Definition 2). The sum of the cut edge weights gives the value of $\delta(S)$. Furthermore, the value of the split generated by $\delta(S)$ is given by the sum of the remaining interactions between agents within the two coalitions C and $\overline{C}$. Thus, the following equivalence applies:

$$v(S) = v(A) - \delta(S), \tag{8}$$

where $S = \{C, \overline{C}\}$, $\forall C, \overline{C} \subseteq A$, s.t. $\overline{C} = A - C$. As a consequence, we can write the value of a cut as the difference between the value of A and S:

$$\delta(S) = v(A) - v(S). \tag{9}$$

Let S be the partition that minimizes $\delta(S)$, i.e., $\delta(S) \leq \delta(S')\ \forall S'$, then the following inequality always holds:

$$v(A) - \delta(S) \geq v(A) - \delta(S') \implies v(S) \geq v(S'). \tag{10}$$

Therefore, finding the partition S which provides minimum value for $\delta(S)$ is equivalent to finding the optimal split $S = \{C, \overline{C}\}$ which maximizes $v(S)$.

This work considers the generic case for ISGs, assuming a complete graph (i.e., fully connected) with positive and negative weights. In this case, the *min-cut* problem is proven to be NP-hard [7] and requires an exponential number of steps with respect to the number of input nodes/agents [9,8].

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_11

4.2 QUBO formulation for Optimal Split

In this section, the *min-cut* problem (or, equivalently, the optimal split problem) is reformulated as QUBO suitable to be executed on a real quantum annealer.

Let $G(A, w)$ be a weighted undirected graph. The *min-cut* (cf. Definition 3) can be formulated as a quadratic objective function of binary variables $\{x_i\}_{i=1,\dots,n}$:

$$\arg\min_{x} \sum_{i=1}^{n} w_i x_i + \sum_{1 \leq i < j \leq n} w_{i,j} x_i (1 - x_j) = x^t W x, \tag{11}$$

where $w_{i,j}$ is the edge weight connecting the nodes i and j, w_i is a self-loop on node i, and the value x_i indicates the membership of one of the two disjointed sets generated by the cut. Therefore, the value of the binary string solution allows differentiating the vertices belonging to the subsets C or $\overline{C}$, as follows:

$$\forall x_i \in A, \quad x_i = \begin{cases} 1 & \text{if } x_i \in C \\ 0 & \text{if } x_i \in \overline{C} \end{cases} \tag{12}$$

where $C \cap \overline{C} = \emptyset$ and $C \cup \overline{C} = A$. Given the QUBO formulation in Eq. (11), we can easily define the corresponding optimization problem of an Ising Hamiltonian with the assignment $x_i \to (1 - Z_i)/2$, where Z_i is the Pauli-Z operator.

4.3 GCS-Q Algorithm

The algorithm follows the strategy of divisive hierarchical clustering (DHC) and applies it in the context of ISGs. Traditional DHC algorithms create a sequence of partitions of a set of n elements, such that an optimality criterion that considers the separation between groups is maximized. In particular, DHCs start with all the elements in a unique group and generate a bipartition to minimize the distance between intra-cluster elements at each step. The number of possible bipartitions of a set of n elements is $\mathcal{O}(2^n)$, and the complete exploration is highly time-consuming. For this reason, standard DHC methods split up the groups according to a distance matrix that explores n^2 different possibilities at each step. This approach allows obtaining a method that scales polynomially ($\mathcal{O}(n^3)$) in the number of elements by drastically reducing the number of partitions examined.

The idea behind GCS-Q is to adopt the top-down strategy of DHC for solving any ISG problem. The algorithm starts initializing the current optimal coalition structure CS^* with the grand coalition g_c where all the agents belong to a single coalition, i.e., $CS^* = \{g_c\}$. Thus, the optimal split problem for g_c is formulated in terms of *min-cut* (Section 4.2) and solved using a quantum annealer. In particular, the annealer evaluates all possible bipartitions of g_c and provides the binary encoding of the bipartition $\{C, \overline{C}\}$ that maximizes the characteristic function (Eq. (3)). Then, the coalition value of g_c is compared with the value of the coalition structure comprising C and $\overline{C}$, i.e.,

$$v(CS^*) = v(\{g_c\}) > v(\{C, \overline{C}\}) = v(C) + v(\overline{C}). \tag{13}$$

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_11

If the splitting produces a lower coalition value, g_c is returned as the optimal coalition structure and the algorithm stops. Otherwise, if the inequality (13) does not hold, the coalition structure $\{C, \overline{C}\}$ is assigned to CS^*. The second step consists of the optimal bipartition for each coalition in the current coalition structure, and the splitting is decided based on the criterion of Eq. (13). This approach allows to generate, at each step and for each coalition , a partition of the agents that has a higher coalition value for the characteristic function. This process continues until none of the coalitions in CS^* can be split to produce a bipartition with a higher coalition value.

Notice that the condition in Eq. (13) provides an automatic stopping criterion for GCS-Q: the algorithm stops if there is no advantage in splitting the coalitions in CS^*. This is a key difference between standard DHC algorithms that always start from a single group and end up with singletons. Furthermore, the additive nature of the coalition value function in Eq. (1) for ISGs allows acting independently on the coalitions to maximize the value of the coalition structure. The pseudocode of GCS-Q is shown in Algorithm 1.

Algorithm 1 GCS-Q Algorithm

Input: Coalition game (A, v), with underlying graph $G(A, w)$ where $|A| = n$ and $w : A \times A \to \mathbb{R}$
Output: Optimal coalition structure CS^*

$CS^* \leftarrow \{\}$ ▷ initialize CS^* with empty list
$queue \leftarrow g_c$ ▷ initialize $queue$ with grand coalition g_c
while $queue \neq \emptyset$ **do**
  $S \leftarrow queue.pop$ ▷ Fetch a coalition from $queue$
  Begin Optimal Split problem
    Create a weight matrix W from edges in S ▷ Eq. (11)
    Define the Ising Hamiltonian $\mathcal{H}$ for W
    Solve Ising Hamiltonian $\mathcal{H}$ on a quantum annealer
    Decode binary string to get C and $\overline{C}$
  End Optimal Split problem
  if $\overline{C} = \emptyset$ **then** ▷ no further splitting of S is inefficient
    add C to CS^*
  else ▷ try further splitting C and $\overline{C}$
    add C to $queue$
    add $\overline{C}$ to $queue$
  end if
end while
Return CS^*

4.4 Discussion

In terms of runtime, the execution of GCS-Q using the classical computation would be $\mathcal{O}(n2^n)$. In particular, given an n-agent ISG, in the case of a superadditive game, the algorithm needs to solve n times the *min-cut* problem for a fully connected graph with positive and negative edge weights, which is NP-hard [9]. Nevertheless, we experimentally show that delegating the problem of finding the

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_11

best bipartition on quantum annealing allows achieving a runtime that scales in the order of n^2. This represents a significant improvement with respect to state-of-the-art quantum and classical algorithms.

The top-down strategy adopted by GCS-Q allows for obtaining several convenient properties: *i)* for an n-agent ISG, GCS-Q always converges in at most n steps. In fact, the algorithm proceeds top-down, starting from a coalition structure containing all the agents into a single coalition (the grand coalition), and it terminates (in the worst case) with a coalition structure containing the singletons. Therefore, the hierarchy is built in $n-1$ steps; *ii)* The proposed algorithm is *anytime*: the splitting in terms of *min-cut* ensures that each coalition is split into disjoint sets, which correspond to two separate coalitions. This approach automatically considers the underlying graph of the coalition game while guaranteeing a valid solution at each time step. *iii)* for superadditive games, GCS-Q always returns the best coalition structure (the grand coalition). Thanks to the top-down approach, the algorithm is initialized with the grand coalition, which has no subpartitions with higher coalition value; *iv)* GCS-Q is an approximate solver which explores a larger portion of the solution space compared to existing clustering-based approaches, such as C-Link [6]. Standard hierarchical clustering algorithms (agglomerative or divisive) examine n^2 bipartitions at each step, leading to a time complexity cubic in the number of elements at the cost of drastically reducing the number of partitions explored. In contrast, GCS-Q evaluates all possible 2^n bipartitions and selects the optimal one. This translates into exploring $\mathcal{O}(2^n)$ possible configurations for the coalition structure at each step. This approach is usually avoided because extremely time-consuming. However, the reformulation as a QUBO allows leveraging a quantum annealer that experimentally shows to solve the problem with an average runtime in the order of n (Section 5.2). Figure 1 provides a small example of how the GCS-Q works.

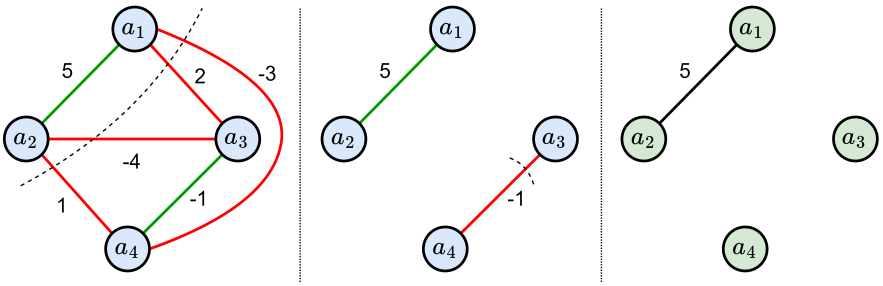

Fig. 1: Graphical representation of how the GCS-Q algorithm works. The figure represents the set of agents alongside the two steps of GCS-Q. The blue nodes indicate the vertices affected by the current min-cut. The green nodes indicate the nodes identified as belonging to the best possible coalition, and further splitting is inefficient. The red edges are the ones that are cut. The green edge denotes the ones which remain after the current optimal cut. Notice that, at each step, the algorithm returns a valid sub-optimal coalition structure.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_11

5 Evaluation

In this section, we comparatively evaluate the GCS-Q in terms of approximation ratio and runtime on synthetic benchmark datasets.

5.1 Experimental Settings

Dataset Generation The standard approach to evaluate algorithms for cooperative games is to generate the values of the coalition function from several probability distributions. In case of ISGs, the datasets refer to the interaction score associated with the pairs of agents. Given n agents, a real value is assigned for each pair of agents, i.e., a total $\binom{n}{2}$ values (unlike standard CSGs where 2^n values are independently generated). The coalitions of size greater than two are assigned according to Eq. (1)[3]. Specifically, the edge weights of the graph game $G(A, w)$ are drawn from the following two distributions:

$$w_{i,j} \sim \mathcal{L}(\mu_1, b) \qquad\qquad w_{i,j} \sim \mathcal{N}(\mu_2, \sigma)$$

where $\mathcal{L}$, $\mathcal{N}$ are the *Laplace* and *Normal* distribution respectively, with $\mu_1 = \mu_2 = 0$ and $\sigma = b = 5$. The parameters are chosen to guarantee positive and negative values for $w_{i,j}$.

Metrics One of the requirements to estimate apriori the approximation error of a CSG algorithm is to assume the value of any coalition to be positive [14]. Since we consider more generic coalition games where the value of a coalition can also be negative, to assess the approximation ratio of GCS-Q we evaluate its performance on a standard benchmark dataset. Given a n-agent coalition game (A, v), the approximation error is defined as the relative error between the value of the *optimal* coalition structure $CS^{(\text{opt})}$ and the value of a *near-optimal* solution $CS^{(\text{n-opt})}$ returned by a selected approximate solver:

$$e_r = \frac{|v(CS^{(\text{opt})}) - v(CS^{(\text{n-opt})})|}{v(CS^{(\text{opt})})} \in [0, 1]. \tag{14}$$

Implementation To find the optimal coalition structure, we run a python implementation of the IDP algorithm [12], which always returns the best possible coalition structure. However, this algorithm runs out of resources with more than 20 agents. Furthermore, rather than running the GCS-Q directly on a real quantum device prone to errors, we evaluate its approximation error using classical computation. In particular, we implement the Algorithm 1, with the optimal split problem solved using a brute force strategy that explores all possible splits. We refer to this approach as GCS-Q$^{(c)}$. In addition, we implement GCS-Q by leveraging the D-Wave 2000Q to solve the sub-task of the optimal split problem. We refer to this approach as GCS-Q$^{(q)}$.

[3] Notice that to better fit standard graph games studied in the literature, we omit self-loop, which would require sampling other n values. However, the GCS-Q formulation is suitable for dealing with graphs containing self-loop.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_11

5.2 Results

GCS-Q Runtime In order to estimate the runtime of the quantum annealing in solving the optimal split problem, we generate 64 ISGs, each with a different number of agents for both distributions (from 2 to 65). Thus, the time-to-solution is recorded for five different runs to obtain an estimation (alongside a variability measure) of the runtime. Results are reported in Figure 2.

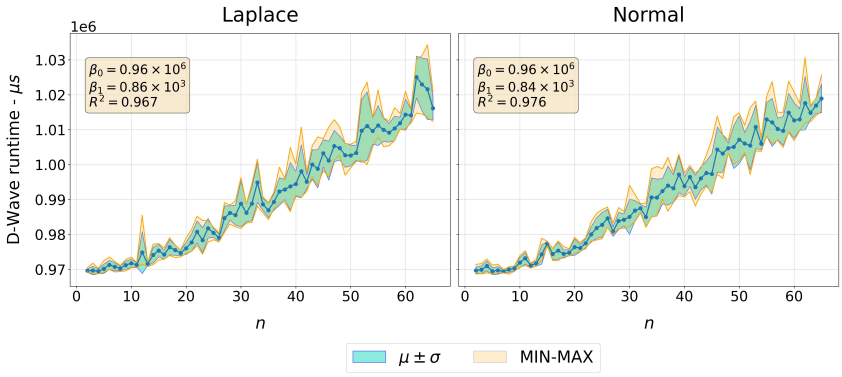

Fig. 2: Runtime of the D-Wave 2000Q when solving the optimal split problem. The same QUBO problem is solved 5 times for each number of agents. The blue line is the average runtime of the 5 experiments. The yellow shaded area represents the maximum and minimum runtimes. The green shaded area is calculated considering the mean and the standard deviation of the runtimes for each problem instance. The runtimes are reported in microseconds (μs).

The order of runtime growth when increasing the number of agents is linear. To confirm this hypothesis, we estimate a linear regression model of the form $T = \beta_0 + \beta_1 n$ where T is the runtime of the quantum annealing, n is the number of agents, and β_0, β_1 are the parameters to be estimated from data. To assess the quality of the linear fitting, we calculate the coefficient of determination R^2, which is equal to 1 in the case of a perfect deterministic linear function between n and T. A value of 97% for both distributions indicates that the relationship between T and n increases linearly. Thus, we can conclude that the average-case complexity of the quantum annealing solution for solving the optimal split problem is linear in the number of agents, i.e., $\Theta(n)$.

Furthermore, to test the efficiency of GCS-Q$^{(q)}$ against GCS-Q$^{(c)}$, we run them on the two benchmark datasets described in Section 5.1 considering games up to 27 agents. Results in terms of runtime are depicted in Figure 3. As expected, GCS-Q$^{(q)}$ runs polynomially in the number of agents, providing a practical quantum advantage over its classical counterpart.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_11

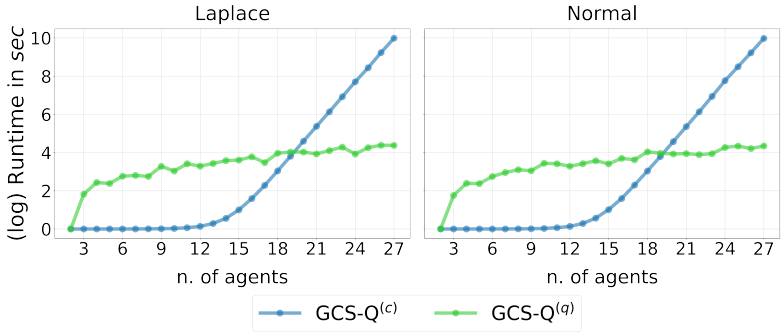

Fig. 3: Runtime of GCS-Q$^{(c)}$ and GCS-Q$^{(q)}$.

Quality Assessment of GCS-Q Although the advantage in terms of runtime, the solution provided by the quantum annealing degrades rapidly due to the limited precision of the quantum device in use. We perform experiments on the two distributions mentioned in Section 5.1 with the number of agents up to 20. In particular, we compare the quality of the solutions obtained with GCS-Q$^{(c)}$ and GCS-Q$^{(q)}$ with best possible coalition structure calculated running IDP. Results are shown in Figure 4.

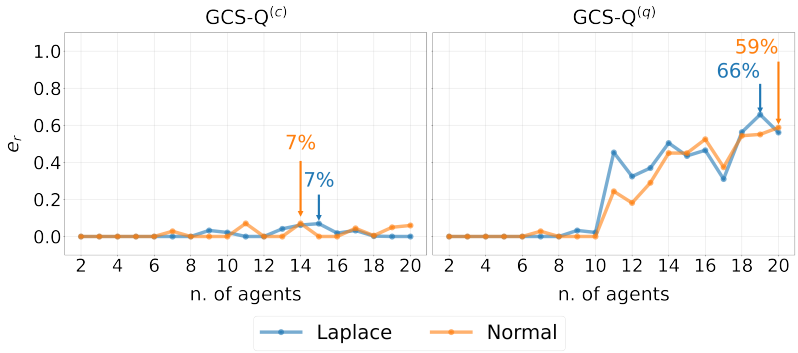

Fig. 4: Assessment of GSC-Q quality in terms of approximation error calculated on the two benchmark datasets described in Section 5.1.

For both distributions, the GCS-Q$^{(c)}$ has a worst-case approximation ratio of 7%. However, the GCS-Q$^{(q)}$ produces the expected solution of the Algorithm 1 only for coalition games of size up to 10. In particular, the approximation error suddenly increases for problems with more than 10 agents, which results in a worst-case approximation error of 66% and 59%). The deterioration is also observed for experiments up to 27 agents. In this case, we consider as baseline $CS^{(\text{opt})}$ the coalition structure returned by the GCS-Q$^{(c)}$ and as approximate

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_11

solution $CS^{(\text{n-opt})}$ the one returned by GCS-Q$^{(q)}$. The results of the quality assessments are reported in Figure 5. The tendency for a decrease in performance is due to the quality of the quantum annealer when solving the optimal split problem. In fact, with $n \geq 11$ the error is cascaded through further executions of the algorithm and the final solution is far from optimality

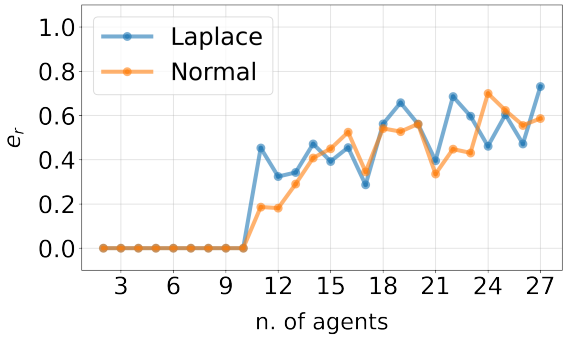

Fig. 5: Relative approximation error GCS-Q$^{(q)}$ algorithm using as baseline the GCS-Q$^{(c)}$.

The limitations of the D-Wave 2000Q have already been emphasized for specific optimization tasks[19]. Nonetheless, the latest generation of D-Wave QPUs, named *Advantage*, outperforms D-Wave 2000Q for any problem size. Furthermore, *Advantage* systems can solve larger problems with up to 120 logical qubits. In some cases, not only the *Advantage* system can find better-quality solutions but it also can find same quality solutions faster [15][4]. However, each problem Hamiltonian must be explicitly studied on the quantum device in use. In this regard, several technical optimization strategies are still possible on the D-Wave 2000Q. For instance, adopting different embedding strategies or leveraging hybrid quantum annealing computation that can deliver better quality solutions for large-size problems at the cost of worsening the runtime (for more details, see D-Wave documentation[5]).

5.3 Performance analysis

In the case of subadditive games, the optimal coalition structure is given by the singletons. This is the worst-case scenario for GCS-Q, which needs to solve the optimal split problem n times. Therefore, considering a linear runtime of the quantum annealing (Sec. 5.2), we obtain a quantum-supported solver which scales quadratically with respect to the number of agents n.

[4] A comparison between Advantage with the D-Wave 2000Q is reported [10]
[5] https://docs.dwavesys.com/docs/latest/index.html

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_11

Thus, we compare this runtime with state-of-the-art classical and quantum solutions when considering ISGs with an underlying fully connected graph. Since ISGs are a special case of general coalition games, we also consider the classic solvers for CSG. The best classical exact solvers for any generic CSG problem are represented by methods based on IDP [12], such as BOSS[5] and DyCE[18], having worst-case complexity of $\mathcal{O}(3^n)$. In the context of ISGs, the CFSS [4] and KGC algorithms [3] have shown excellent results with sparse graphs, but the worst-case complexity for a complete graph remains $\mathcal{O}(n^n)$. The best approximate solution in terms of the runtime is C-link [6], which is based on hierarchical agglomerative clustering and has cubic complexity in the number of agents, i.e., $\mathcal{O}(n^3)$. Finally, the only general quantum solution for CSG is BILP-Q[16], which showed a runtime of $\mathcal{O}(2^n)$ using quantum annealing A graphical comparison of state-of-the-art classical and quantum solutions is provided in Figure 6.

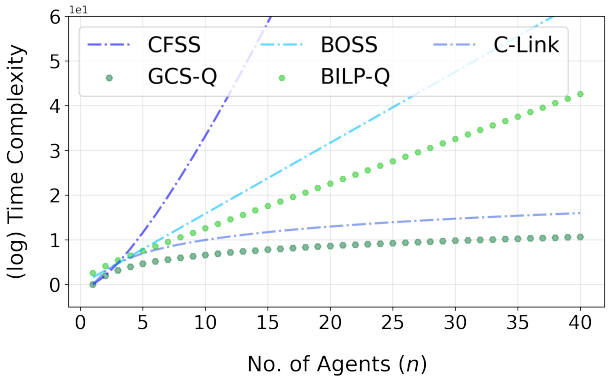

Fig. 6: Cost complexity as a function of the number of agents n. Classical solutions are indicated with blue lines, while quantum solutions are in green.

The ability of GCS-Q to be executed with a runtime quadratic in the number of agents makes it the best solver for ISGs.

6 Conclusion and Future Work

In this work, we proposed GCS-Q, the first quantum-supported solution for coalition structure generation in Induced Subgraph Games. The key idea is to partition the graph underlying the coalition game into two subsets iteratively in order to obtain a coalition structure with a better coalition value. By delegating the task of finding the optimal split to a quantum annealer, we obtain a solver capable of running faster than the state-of-the-art solutions (quantum and classical). Given a n-agent coalition game, the ability of the D-Wave 2000Q quantum device to solve the optimal split problem in linear time with respect to n allows for conveying an overall runtime that scales in the order of n^2. Furthermore, by exploring all possible partitions of a given coalition, the GCS-Q

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_11

examines a larger portion of the solution space compared with other approximate solvers, such as C-Link[6]. In fact, this latter adopts a bottom-up strategy and never considers the global distribution of the agents. Another important feature of GCS-Q is the ability to provide sub-optimal solutions during its execution, which makes it an anytime solver.

In addition, we provided a practical implementation of GCS-Q and evaluated its performance on standard benchmark datasets. Specifically, we generated coalition games with fully connected graphs, sampling the edge weights of the graph underlying the coalition games from two distributions (*Laplace* and *Normal*). We implemented two variants of the GCS-Q (Algorithm 1): GCS-Q$^{(c)}$ is executed entirely on a classical computer and served to estimate the expected approximation ratio (93%) for the benchmark datasets. The second approach, named GCS-Q$^{(q)}$, leverages the D-Wave 2000Q for solving the sub-task of the optimal split. As expected, the GCS-Q$^{(q)}$ scales polynomially in the number of agents and exponentially faster than GCS-Q$^{(c)}$. However, when calculating the quality of the solutions, the performance deteriorates due to the limitations of the quantum hardware in use. For this reason, the main challenge to tackle in the near future is the investigation of alternative embedding strategies or the adoption of hybrid quantum-classical solvers proposed by D-Wave.

Another natural follow-up is the execution of GCS-Q on better quantum hardware, such as the D-Wave *Advantage*. This latest generation of quantum devices has outperformed the D-Wave 2000Q in terms of both quality of solutions obtained and runtime in several combinatorial optimization problems.

In conclusion, we showed the feasibility and the benefit of adopting quantum computation in multi-agent systems with a novel quantum-supported solution suitable for solving practical real-world AI problems.

Acknowledgments

This work has been funded by the German Ministry for Education and Research (BMB+F) in the project QAI2-QAICO under grant 13N15586.

Code Availability

All code to generate the data, figures, analyses and additional technical details on the experiments are available at https://github.com/supreethmv/GCS-Q.

References

1. Aziz, H., De Keijzer, B.: Complexity of coalition structure generation. arXiv preprint arXiv:1101.1007 (2011)
2. Bachrach, Y., Kohli, P., Kolmogorov, V., Zadimoghaddam, M.: Optimal coalition structure generation in cooperative graph games. In: Twenty-Seventh AAAI Conference on Artificial Intelligence (2013)
3. Bistaffa, F., Chalkiadakis, G., Farinelli, A.: Efficient coalition structure generation via approximately equivalent induced subgraph games. IEEE Transactions on Cybernetics (2021)

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_11

4. Bistaffa, F., Farinelli, A., Cerquides, J., Rodriguez-Aguilar, J.A., Ramchurn, S.D.: Anytime coalition structure generation on synergy graphs. In: 13th International Conference on Autonomous Agents and Multi-Agent Systems (05/05/14 - 09/05/14) (2014)
5. Changder, N., Aknine, S., Ramchurn, S.D., Dutta, A.: Boss: A bi-directional search technique for optimal coalition structure generation with minimal overlapping. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 15765–15766 (2021)
6. Farinelli, A., Bicego, M., Ramchurn, S., Zuchelli, M.: C-link: a hierarchical clustering approach to large-scale near-optimal coalition formation. In: 23rd International Joint Conference on Artificial Intelligence (IJCAI 2013) (03/08/13 - 09/08/13) (2013)
7. Garey, M.R., Johnson, D.S.: Computers and intractability, vol. 174. freeman San Francisco (1979)
8. González-Arangüena, E., Manuel, C.M., del Pozo, M.: Values of games with weighted graphs. European Journal of Operational Research **243**(1), 248–257 (2015)
9. MCCORMICK, S.T., Rao, M., RINALDI, G.: When is min cut with negative edges easy to solve? Easy and difficult objective functions for max cut. CONSIGLIO NAZIONALE DELLE RICERCHE (CNR) (2000)
10. McGeoch, C., Farré, P.: The d-wave advantage system: An overview. D-Wave Systems Inc., Burnaby, BC, Canada, Tech. Rep (2020)
11. Rahwan, T., Jennings, N.R.: Coalition structure generation: Dynamic programming meets anytime optimization. In: AAAI Conference on Artificial Intelligence (2008)
12. Rahwan, T., Jennings, N.R.: An improved dynamic programming algorithm for coalition structure generation. In: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 3. pp. 1417–1420 (2008)
13. Rahwan, T., Ramchurn, S.D., Dang, V.D., Giovannucci, A., Jennings, N.R.: Anytime optimal coalition structure generation. In: AAAI. vol. 7, pp. 1184–1190 (2007)
14. Sandholm, T., Larson, K., Andersson, M., Shehory, O., Tohmé, F.: Coalition structure generation with worst case guarantees. Artificial intelligence **111**(1-2), 209–238 (1999)
15. Tasseff, B., Albash, T., Morrell, Z., Vuffray, M., Lokhov, A.Y., Misra, S., Coffrin, C.: On the emerging potential of quantum annealing hardware for combinatorial optimization. arXiv preprint arXiv:2210.04291 (2022)
16. Venkatesh, S.M., Macaluso, A., Klusch, M.: Bilp-q: quantum coalition structure generation. In: Proceedings of the 19th ACM International Conference on Computing Frontiers. pp. 189–192 (2022)
17. Voice, T., Polukarov, M., Jennings, N.R.: Coalition structure generation over graphs. Journal of Artificial Intelligence Research **45**, 165–196 (2012)
18. Voice, T., Ramchurn, S.D., Jennings, N.R.: On coalition formation with sparse synergies. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 1. p. 223–230. International Foundation for Autonomous Agents and Multiagent Systems (2012)
19. Willsch, D., Willsch, M., Gonzalez Calaza, C.D., Jin, F., De Raedt, H., Svensson, M., Michielsen, K.: Benchmarking advantage and d-wave 2000q quantum annealers with exact cover problems. Quantum Information Processing **21**(4), 1–22 (2022)

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_11