QuACS : Variational Quantum Algorithm for Coalition Structure Generation in Induced Subgraph Games

Supreeth Mysore Venkatesh
supreeth.mysore@dfki.de
German Research Center for Artificial Intelligence (DFKI)
Saarbruecken, Germany

Antonio Macaluso
antonio.macaluso@dfki.de
German Research Center for Artificial Intelligence (DFKI)
Saarbruecken, Germany

Matthias Klusch
matthias.klusch@dfki.de
German Research Center for Artificial Intelligence (DFKI)
Saarbruecken, Germany

ABSTRACT

Coalition Structure Generation (CSG) is an NP-Hard problem in which agents are partitioned into mutually exclusive groups to maximize their social welfare. In this work, we propose QuACS, a novel hybrid quantum-classical algorithm for Coalition Structure Generation in Induced Subgraph Games (ISGs). Starting from a coalition structure where all the agents belong to a single coalition, QuACS recursively identifies the optimal partition into two disjoint subsets. This problem is reformulated as a QUBO and then solved using QAOA. Given an n-agent ISG, we show that the proposed algorithm outperforms existing approximate classical solvers with a runtime of $O(n^2)$ and an expected approximation ratio of 92%. Furthermore, it requires a significantly lower number of qubits and allows experiments on medium-sized problems compared to existing quantum solutions. To show the effectiveness of QuACS we perform experiments on standard benchmark datasets using quantum simulation.

CCS CONCEPTS

• **Computing methodologies** → **Intelligent agents**; • **Theory of computation** → **Quantum information theory**.

KEYWORDS

Quantum AI, Quantum Computing, Coalition Game Theory

ACM Reference Format:
Supreeth Mysore Venkatesh, Antonio Macaluso, and Matthias Klusch. 2023. QuACS : Variational Quantum Algorithm for Coalition Structure Generation in Induced Subgraph Games. In *20th ACM International Conference on Computing Frontiers (CF '23), May 9–11, 2023, Bologna, Italy*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3587135.3592192

1 INTRODUCTION

A generic coalition game, also called a *Characteristic Function Game (CFG)* (A, v) comprises a set of intelligent agents A, and a characteristic function $v : \mathcal{P}(A) \rightarrow \mathbb{R}$ which maps every non-empty subset (*coalition*) of A to a real value. In many practical applications, there are constraints that may limit the formation of coalitions and the synergies between agents can be expressed as a graph[6]. In this case, the value of a coalition only depends on the pairwise interactions connecting the agent in the graph, and such problems are usually referred to as CFG is *Induced Subgraph Game (ISG)* [6].

Given an ISG (A, v) represented as a connected, undirected, weighted graph $G(A, w)$ where the nodes represent the agents $A = \{a_1, a_2, ...a_n\}$ and the edge weights w_{ij} denote the synergy between a_i and a_j for $i, j \in \{1, \ldots, n\}$, the value of the characteristic function for a coalition C can be expressed as $v(C) = \sum_{i,j \in C} w_{ij}, \forall C \subseteq A$. A Coalition Structure (CS) is a complete set partition of A consisting of a set of coalitions $\{C_1, C_2, \ldots, C_k\}$ such that $\bigcup_{i=1}^{k} C_i = A$ and $C_i \cap C_j = \emptyset$ for any $i, j \in \{1, 2, \ldots, k\}$ and $i \neq j$. The goal is to find the optimal coalition structure CS^* such that:

$$CS^* = \arg\max_{CS} \sum_{C \in CS} \sum_{i,j \in C} w_{i,j} \tag{1}$$

In the scope of this paper, we refer to the problem of CSG in ISG as the *ISG problem* and we assume a fully connected graph allowing both positive and negative edge weights. In this respect, the ISG problem remains NP-Hard [1]. Notice that graph-restricted games, such as ISGs, have realistic use cases, e.g., social network analysis [10] to discover collective groups of people with similar interests, or content downloading in self-driving cars [13].

2 RELATED WORKS

ISG problems can be seen as special cases of standard CFGs and therefore, any algorithm to solve the generic CSG problem for CFGs can be applied in the context of ISGs (not vice-versa). The time complexity of state-of-the-art exact solvers for generic CSGs [4, 14, 18] scales as $O(3^n)$ where n is the number of agents.

In order to deal with this exponential complexity, approximate solvers, such as C-link [9] can be adopted. The C-Link algorithm is inspired by agglomerative clustering which follow a greedy bottom-up approach to determine the profit of merging (only) two coalitions at each step. This approach leads to an overall worst-case time complexity of $O(n^3)$ for a coalition game with n agents at the cost of drastically reducing the exploration of the solution space. This leads to a worst-case approximation ratio of 80% with respect to the quality of the solution. Specifically for ISGs, CFSS [3] is an anytime exact solver and the state-of-the-art solver *k-constrained Graph Clustering (KGC)* [2] which is an *Integer Linear Programming (ILP)* based solver. Although both CFSS and KGC perform well for sparse graphs, they might end up exploring all possible solutions for complete graphs with a complexity of $O(n^n)$.

Recently, a few quantum computing solutions have been proposed for solving the CSG problem. BILP-Q [16] is the first general

CF '23, May 9–11, 2023, Bologna, Italy

ACM ISBN 979-8-4007-0140-5/23/05.
https://doi.org/10.1145/3587135.3592192

quantum algorithm for any CSG problem. It operates by reformulating the problem as a Quadratic Binary Unconstrained Optimization (QUBO) problem and solves it using both gate-based quantum computing and quantum annealing. Although BILP-Q potentially outperforms state-of-the-art classical solutions for generic CFGs, it requires the number of logical qubits to be $O(2^n)$, which is a significant limitation considering near-term quantum technology.

In addition, two possible quantum annealing solvers for ISGs have been proposed. GCS-Q [17] is an anytime approximate solver for any generic ISG which follows a top-down approach to find a near-optimal coalition structure. Particularly, it performs multiple calls to a D-Wave quantum annealer to find the optimal split to split for a given coalition. Alternatively, other existing approaches [12] map the graph of an ISG into specific quantum annealer hardware architecture to find the optimal coalition structure. However, this method cannot be adopted for any problem instance as the mapping is dependent on the hardware specifications.

In this work, we propose QuACS (Quantum Algorithm for Coalition Structure generation), a novel hybrid quantum-classical algorithm for ISG problems. QuACS adopts the same top-down approach proposed by GCS-Q [17], where the coalitions are recursively split into two disjoint subsets but uses gate-based quantum optimization (QAOA [8]) for finding the optimal bipartition at each step. This allows investigating its scalability and usability in terms of runtime, number of gates, and number of qubits, to properly compare it with respect to existing classical and quantum solvers. Starting with a coalition structure containing a single coalition including all the agents, QuACS runs until no further split provides better coalition value for any coalition in the current optimal coalition structure. Thanks to this convenient strategy, the proposed solution is anytime and allows for obtaining near-optimal solutions in polynomial time. These two features are essential for real-world situations where a fast near-optimal solution is needed. We show that QuACS outperforms existing approximate solvers both in terms of runtime and quality of the solution. Furthermore, it requires a significantly lower number of qubits with respect to BILP-Q [16].

As a second contribution, we implement QuACS using quantum simulation on standard benchmark datasets with a number of agents up to 20. We will show that the proposed approach performs well even for shallow-depth QAOA ($p = 1$) with a worst-case approximation ratio of approximately 80%. Also, the performances improve when tuning p up to 92%.

3 ALGORITHM

Given an n-agent ISG (cf. Definition 1) with a fully connected underlying graph, QuACS initially assigns all the agents to a single coalition, the grand coalition g_c. Thus the algorithm considers all possible splits of g_c into two disjoint coalitions, evaluating the correspondent value of the characteristic function. If none of the generated bipartitions provides a value greater than that of $v(g_c)$, then the algorithm terminates by returning g_c as the best coalition structure CS^* (e.g., in the case of superadditive games). Otherwise, the optimal bipartition $\{C, \overline{C}\}$ of g_c, which maximizes the characteristic function, is selected, and the optimal coalition structure CS^* becomes $\{C, \overline{C}\}$, where the two sets of agents act independently from each other. The process of finding the optimal bipartition is then repeated for each coalition in the current optimal coalition structure CS^*, and the algorithm proceeds until none of the coalitions in CS^* can be split in a way to provide a better coalition value. In particular, the algorithm terminates when

$$v(S) > v(C) + v(\overline{C}) \qquad \forall S \in CS^* \tag{2}$$

where $C \cup \overline{C} = S$, $C \cap \overline{C} = \emptyset$. To find an optimal split (or optimal bipartition), QuACS divides the underlying connected graph of a coalition into two disconnected subgraphs by removing the edges that maximize the sum of the remaining edge weights in the subgraphs. In other words, the nodes of the underlying graph are separated into two mutually exclusive subsets such that the sum of the edge weights in the subgraphs induced by the subset of vertices is maximum. Thus, finding the optimal split for a given ISG is equivalent to performing the weighted minimum cut (min-cut) in the underlying graph [6], which minimizes the sum of edge weights that are removed while partitioning the vertices into two disjoint sets. For arbitrary edge weights, the min-cut is proven to be NP-Hard [11] and the exhaustive enumeration of all possible bipartitions for a coalition of n agents is $O(2^n)$.

In order to improve the runtime for finding the best coalition structure, QuACS leverages the QAOA [8] for solving the min-cut problem at each step. In fact, this problem can be reformulated as a QUBO [5] and the solution can be retrieved using hybrid quantum-classical optimization. Specifically, the solution to the min-cut provided by training the correspondent QAOA is a binary string whose values correspond to each agent belonging to one of the two partitions. Thus, this process of formulating the optimal split as a QUBO and solving it using QAOA is repeated at every step of QuACS to generate the optimal splits.

The pseudocode for QuACS is reported in Algorithm 1. Example 3 shows how QuACS proceeds for a 4-agent game.

Algorithm 1 Outline of QuACS

Require: Set of n agents $A = \{a_1, a_2, ., a_n\}$, weights $w : A \times A \rightarrow \mathbb{R}$
  Initialize $CS^* \leftarrow g_c$ ▷ grand coalition $g_c = \{A\}$
  for an unexplored coalition $S \in CS^*$ **do**
    Derive the Ising Hamiltonian for min-cut of S
    Solve Ising Hamiltonian using QAOA
    Decode binary string to get $C, \overline{C}$ ▷ where $C \cup \overline{C} = S$, $C \cap \overline{C} = \emptyset$
    if $v(C) + v(\overline{C}) \geq v(S)$ **then**
      Remove S
      Add $C, \overline{C}$ to CS^*
    end if
  end for
Ensure: Optimal Coalition Structure CS^*

Example. Given an ISG with four agents $\{a_1, a_2, a_3, a_4\}$ and the edge weights between them defined as[1]:

$$w_{12} = 2,\ w_{13} = 6,\ w_{14} = -4,\ w_{23} = -5,\ w_{24} = -1,\ w_{34} = 1$$

The QuACS finds the optimal coalition structure CS^* as follows. At each step, the split with the highest value is chosen. CS^* always

[1]We consider a simple example with all agents/nodes having no self-loop, i.e., their utility is equal to zero when working separately.

contains the best solution found until then and its value is given by $v(CS^*)$. The algorithm terminates when none of the coalitions in CS^* has a better split. Here, $\{\{a_1, a_3\}, \{a_2\}, \{a_4\}\}$ is the best way to partition the agents in A.

Step	Coalition (S)	Values to Compare	CS^*	$v(CS^*)$
0	-	-	$\{\{a_1, a_2, a_3, a_4\}\}$	-1
1	$\{a_1, a_2, a_3, a_4\}$	$v(\{a_1, a_2, a_3, a_4\}) = -1$ $v(\{a_1\}) + v(\{a_2, a_3, a_4\}) = -5$ $v(\{a_1, a_2\}) + v(\{a_3, a_4\}) = 3$ $v(\{a_2\}) + v(\{a_1, a_3, a_4\}) = 3$ $v(\{a_1, a_3\}) + v(\{a_2, a_4\}) = 5$ $v(\{a_3\}) + v(\{a_1, a_2, a_4\}) = -3$ $v(\{a_1, a_4\}) + v(\{a_2, a_3\}) = -9$ $v(\{a_4\}) + v(\{a_1, a_2, a_3\}) = 3$	$\{\{a_1, a_3\}, \{a_2, a_4\}\}$	5
2	$\{a_1, a_3\}$	$v(\{a_1, a_3\}) = 6$ $v(\{a_1\}) + v(\{a_3\}) = 0$	$\{\{a_2.a_4\}, \{a_1, a_3\}\}$	5
3	$\{a_2, a_4\}$	$v(\{a_2, a_4\}) = -1$ $v(\{a_2\}) + v(\{a_4\}) = 0$	$\{\{a_1, a_3\}, \{a_2\}, \{a_4\}\}$	6

Table 1: The table illustrates the working of QuACS for an ISG with four agents. The green box highlights the splits of S with the maximum value chosen at each step.

3.1 Performance Analysis

In this section, we analyze the performance of QuACS in terms of the number of qubits, number of gates, and runtime.

Lemma 3.1. *QuACS that uses a p-layered QAOA circuit, requires $O(n)$ qubits to solve an n-agent ISG problem.*

Proof. In the execution of QuACS, the task of finding the optimal split is reduced to the min-cut problem, reformulated as QUBO, and solved using QAOA. The number of qubits required to solve a QUBO matrix of size $n \times n$ is equal to n. For a given n-agent problem, the largest QUBO to be solved is the one corresponding to the first step of QuACS where the grand coalition has to be split. Given the top-down approach of QuACS , any further execution of the QAOA operates on coalitions whose size is lower than n, which means that the number of qubits required is strictly lower than n. Thus, the qubit complexity of QuACS is $O(n)$. □

Lemma 3.2. *QuACS that uses a p-layered QAOA circuit, requires $O(n^2 p)$ single and/or two qubit gates to solve an n-agent ISG problem.*

Proof. According to the Lemma 3.1, the largest instance of QAOA in QuACS requires n qubits. In this case, the first step of QAOA generates an equal superposition of 2^n possible states through the use of n Hadamard gates. Then, for each non-zero interaction in the QUBO matrix of the min-cut (cost Hamiltonian H_c), three gates (two CNOT gates and a local single-qubit R_Z gate) are used, plus an additional R_Z applied to each qubit. Notice that, in the case of a fully-connected graph, all the off-diagonal elements of the QUBO matrix are diverse from zero. The number of the off-diagonal elements of the QUBO is $n(n-1)/2$, which can be approximated as n^2. Finally, n Pauli-X single-qubit rotation gates R_X are applied (mixing Hamiltonian H_B). For a p-layered QAOA[2], the whole set of gates is repeated p times (except for the Hadamard) Thus, the total number of single or two-qubit of QuACS is $n + p(n^2 + n)$, which can be approximated as $O(n^2 p)$. □

[2]for more details on the QAOA implementation see [5]

In terms of runtime, the best case for QuACS is when none of the splits of the g_c (grand coalition) provide a higher coalition value than g_c. For a generic n-agent ISG, solving the min-cut is NP-Hard[11], i.e., it requires at most $O(2^n)$ operations to evaluate all possible bipartitions of g_c. When the optimal coalition structure is defined by the set of singletons, QuACS has to process $n-1$ times the optimal split, from the grand coalition to the set of singletons. Classically, the overall runtime of QuACS when executed using classical computation is:

$$\sum_{k=2}^{n} O(2^k) = O(n2^n). \tag{3}$$

Instead, using a p-layered QAOA for solving the min-cut provides a runtime of $O(np)$[5], assuming a negligible cost for the optimization process. Therefore, assuming $p = 1$, QuACS implements $O(n)$ times the QAOA with a runtime of $O(n)$, which leads to an overall worst-case runtime of QuACS is $O(n^2)$.

3.2 Evaluation

For the experiments, we generate ISGs with edge weights sampled from Normal and Random (Uniform) distributions centered in 0 for generating both positive and negative values. To assess the quality of QuACS , we implement IDP [14] to find the optimal coalition structure and two variants of QuACS differing from each other for the method adopted for finding the optimal split at each step: $QuACS_c$ solves the min-cut using the classical QUBO solver while $QuACS_q$ uses the QAOA. All the implementation use IBM Qiskit an aer_simulator to perform the experiments. The runtime for both distributions with ISGs up to 20 agents, considering a 1-layer QAOA for $QuACS_q$ are reported Figure 1.

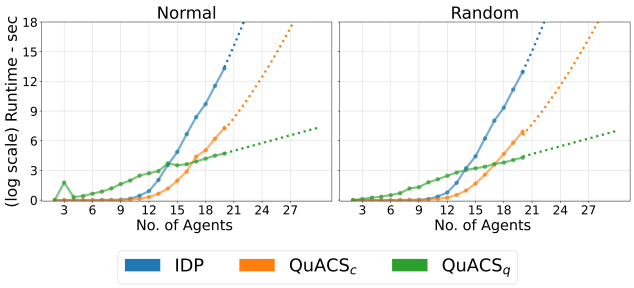

Figure 1: Empirical runtimes of IDP, $QuACS_c$ (Eq. 3) and $QuACS_q$. The plots are analytically extrapolated denoted by dotted lines.

It can be observed that the runtimes of IDP and $QuACS_c$ scale exponentially as expected, while $QuACS_q$ is polynomial to n, which complies with the runtimes discussed in Section 3.1.

Since QuACS adopts a greedy top-down strategy, it is an approximate solver, and the quality of the solutions needs to be analyzed. Theoretical quality analysis of an approximate solver for CSG problem is possible when the game is restricted to allow only positive coalition values [15]. As we consider a more generic problem instance, we can only investigate the approximation empirically. The value of the optimal coalition structure from IDP ($v(CS_E)$) is used as the baseline to evaluate the approximation error in the value obtained from an approximate solver ($v(CS_A)$) using the relation

$Er = \frac{|v(CS_E) - v(CS_A)|}{v(CS_E)} \in [0, 1]$. With IDP as an exact solver, relative errors of $QuACS_c$ and $QuACS_q$, named Er_c and Er_q respectively, are calculated. We compute the quality solution using two different approaches for $QuACS_q$: assuming $p = 1$ and training the QAOA using different values for p, i.e., $p \in [1, 5]$. Results are shown in Figure 2.

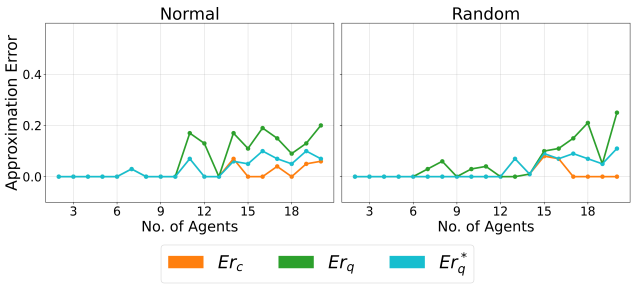

Figure 2: Approximation errors for $QuACS_c$ and $QuACS_q$. Er_q^* indicates the results of $QuACS_q$ when tuning p up to 5.

The minimum value for Er_c is equal to 8% which translates to an expected approximation ratio for QuACS of 92% when solving the min-cut problem classically. When using QAOA, QuACS is more prone to error due to the intrinsically probabilistic nature of quantum simulators and the uncertainty of the optimization process during the training. In this case, for $p = 1$, the maximum Er_q is 7% up to 10 agents and the quality deteriorates as long the problem size increases. When tuning p, the quality of the solution improves, and Er_q^* seems to converge to Er_c.

4 DISCUSSION

In the previous section, we showed that QuACS solves the coalition structure generation for ISGs in polynomial time with an expected worst-case approximation ratio of 92%. QuACS always provides a valid coalition structure (complete set partition of the agents) at any step of the computation since the split operation produces a disjoint partition of the agents. Thus QuACS is an anytime approach. When compared with alternative gate-based quantum solutions (e.g., BILP-Q[16]), which require $O(2^n)$ logical qubits to be implemented, QuACS can be implemented using at most n qubits for an n-agent coalition game. However, unlike BILP-Q, QuACS is an approximate solver, suitable for ISGs only. Classically, the best CSG solver for CFG is BOSS [4], an exact and not anytime solver with a time complexity of $O(3^n)$. The approximate solver C-Link [9] has a time complexity of n^3, with an experimental approximation of 80%.

Thus, QuACS, with a runtime that scales quadratically in the number of agents and an experimental approximation ratio of 92% outperforms existing classical solvers and is suitable for near-term quantum technology.

5 CONCLUSION

In this work, we presented QuACS, an approximate and anytime quantum solver for finding the optimal coalition structure in ISGs. Starting from the set of all agents, QuACS tries to find optimal bipartition iteratively while delegating the optimal split problem (as min-cut) to the QAOA. QuACS scales quadratically in the number of agents n and outperforms existing approximate solvers in terms of both runtime and approximation ratio.

We also implemented QuACS using quantum simulation, showing that the proposed algorithm is already a credible alternative solution for problems with tens of agents. As future work, techniques like warm-starting the QAOA [7] can be adopted for better training the quantum circuits. An additional improvement can be given by parallelizing the task of finding the optimal splits on multiple quantum computers/simulators. Finally, the adoption of QuACS can be used for generic CFGs by finding the corresponding approximately equivalent ISG [2].

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 are publicly available at https://github.com/supreethmv/QuACS.

REFERENCES

[1] Yoram Bachrach, Pushmeet Kohli, Vladimir Kolmogorov, and Morteza Zadimoghaddam. 2013. Optimal coalition structure generation in cooperative graph games. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*.
[2] Filippo Bistaffa, Georgios Chalkiadakis, and Alessandro Farinelli. 2021. Efficient Coalition Structure Generation via Approximately Equivalent Induced Subgraph Games. *IEEE Transactions on Cybernetics* (2021).
[3] Filippo Bistaffa, Alessandro Farinelli, Jesús Cerquides, Juan Antonio Rodriguez-Aguilar, and Sarvapali D Ramchurn. 2014. Anytime coalition structure generation on synergy graphs. (2014).
[4] Narayan Changder, Samir Aknine, Sarvapali D Ramchurn, and Animesh Dutta. 2021. BOSS: A Bi-directional Search Technique for Optimal Coalition Structure Generation with Minimal Overlapping (Student Abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 15765–15766.
[5] Gavin E Crooks. 2018. Performance of the quantum approximate optimization algorithm on the maximum cut problem. *arXiv preprint arXiv:1811.08419* (2018).
[6] Xiaotie Deng and Christos H Papadimitriou. 1994. On the complexity of cooperative solution concepts. *Mathematics of operations research* 19, 2 (1994), 257–266.
[7] Daniel J Egger, Jakub Mareček, and Stefan Woerner. 2021. Warm-starting quantum optimization. *Quantum* 5 (2021), 479.
[8] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. 2014. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028* (2014).
[9] Alessandro Farinelli, Manuele Bicego, Sarvapali Ramchurn, and Marco Zuchelli. 2013. C-link: A hierarchical clustering approach to large-scale near-optimal coalition formation. (2013).
[10] Santo Fortunato. 2010. Community detection in graphs. *Physics reports* 486, 3-5 (2010), 75–174.
[11] Michael R Garey and David S Johnson. 1979. Computers and intractability. *A Guide to the* (1979), 210.
[12] Florin Leon, Andrei-Ştefan Lupu, and Costin Bădică. 2017. Multiagent coalition structure optimization by quantum annealing. In *International Conference on Computational Collective Intelligence*. Springer, 331–341.
[13] Hafez Eslami Manoochehri and Rym Z Wenkstern. 2017. Dynamic coalition structure generation for autonomous connected vehicles. In *2017 IEEE International Conference on Agents (ICA)*. IEEE, 21–26.
[14] Talal Rahwan and Nicholas R Jennings. 2008. 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*. 1417–1420.
[15] Tuomas Sandholm, Kate Larson, Martin Andersson, Onn Shehory, and Fernando Tohmé. 1999. Coalition structure generation with worst case guarantees. *Artificial intelligence* 111, 1-2 (1999), 209–238.
[16] Supreeth Mysore Venkatesh, Antonio Macaluso, and Matthias Klusch. 2022. BILP-Q: quantum coalition structure generation. In *Proceedings of the 19th ACM International Conference on Computing Frontiers*. 189–192.
[17] Supreeth Mysore Venkatesh, Antonio Macaluso, and Matthias Klusch. 2022. GCS-Q: Quantum Graph Coalition Structure Generation. *arXiv preprint arXiv:2212.11372* (2022).
[18] Thomas Voice, Sarvapali Ramchurn, and Nick Jennings. 2012. On coalition formation with sparse synergies. (2012).