Quantum Information Processing (2021) 20:302
https://doi.org/10.1007/s11128-021-03239-1

Quantum k-community detection: algorithm proposals and cross-architectural evaluation

Felix G. Gemeinhardt[1] · Robert Wille[2] · Manuel Wimmer[1]

Received: 20 February 2021 / Accepted: 24 August 2021 / Published online: 9 September 2021

Abstract

Emerging quantum technologies represent a promising alternative for solving hard combinatorial problems in the post-Moore's law era. For practical purposes, however, the current number of qubits limits the direct applicability to larger real-world instances in the near-term future. Therefore, a promising strategy to overcome this issue is represented by hybrid quantum classical algorithms which leverage classical as well as quantum devices. One prominent example of a hard computational problem is the community detection problem: a partition of a graph into distinct communities such that the ratio between intra-community and inter-community connectivity is maximized. In this paper, we explore the current potential of quantum annealing and gate-based quantum technologies to solve the community detection problem for an arbitrary number of communities. For this purpose, existing algorithms are (re-)implemented and new hybrid algorithms, that can be run on gate-model devices, are proposed. Their performance on standardized benchmark graphs has been evaluated and compared to the one of a state-of-the-art classical heuristic algorithm. Although no quantum speed-up has been achieved, the existing quantum annealing-based methods as well as the novel hybrid algorithms for gate-based quantum computers yield modularity values, which are similar to those of the classical heuristic. However, the modular architecture of the used algorithms allows for fast utilization of more powerful quantum technologies once they become available. **Reproducibility:**

✉ Felix G. Gemeinhardt
felix.gemeinhardt@jku.at

Robert Wille
robert.wille@jku.at

Manuel Wimmer
manuel.wimmer@jku.at

[1] Institute for Business Informatics - Software Engineering, CDL-MINT, Johannes Kepler Universität Linz, Linz, Austria

[2] Institute for Integrated Circuits, Software Competence Center Hagenberg GmbH (SCCH), Johannes Kepler Universität Linz, Linz, Austria

Our code and data are publicly available (Github in Quantum Modularization. https://github.com/jku-winse/quantum_modularization 2021).

Keywords Community detection · Quantum computing · Quantum annealing · Modularity · NISQ

1 Introduction

Problem Statement and Related Work: The community detection problem is about determining a partition of a graph into distinct communities such that the ratio between intra-community and inter-community connectivity is maximized. In this context, we refer to k-community detection as maximizing with respect to an arbitrary number of k communities. This problem is highly relevant in many practical applications such as the analysis of social networks [1,11,29], medical science [32], or biology and chemistry [16,27]. In particular in software engineering, community detection is essential. Here, large systems frequently need to be decomposed into smaller ones in order to reduce the complexity toward manageable levels. This process is known as modularization [28]. Unfortunately, community detection or modularization for that matter are highly non-trivial problems in which the number of possibilities increases exponentially, according to the Bell number. Because of that, e.g., for a system with 10 elements there are already 115.975 possible modularizations [4]. Accordingly, several (classical) solutions to this problem have been developed in the past (see, e.g., [3,6,8,12–14,16,19]).

With the dawn of quantum computing technologies, interesting alternatives to these classical solutions emerged and got investigated by researchers and engineers. The two most prominent general approaches that are currently utilized for this purpose are:

- *Gate-based (GB) quantum computing*, where operators are applied to a quantum system to manipulate its state. Within this paradigm, algorithms for combinatorial optimization (e.g., QAOA [10] or Grover Adaptive Search [15]) can be utilized for community detection.
- *Quantum Annealing (QA)*, where the function which has to be optimized is encoded in the Hamiltonian of the quantum system.

These are considered the most promising candidates for demonstrating quantum advantage in solving a computationally hard problem faster than a classical state-of-the-art algorithm [35]. The recent review by Akbar & Saritha [1] provides a comprehensive overview to the field, which got established under the term *quantum community detection*. The review also contains a comparative analysis with a focus on quantum inspired algorithms, like the quantum inspired evolutionary algorithm [18].

However, current implementations of those quantum computing technologies are still in its infancy and are highly limited in its number of qubits. Because of this, the use of *hybrid algorithms* is considered a necessity for practical community detection problems [33]. In those hybrid algorithms (e.g., [30,33,35]), the overall solution is still handled by a classical framework, but adequately formulated sub-problems are solved by a quantum device (followed by the re-integration of the correspondingly determined solutions into the classical framework). As an example, the Quantum Local Search (QLS) algorithm [33] allows for a bipartition of a graph into two communities, where the manageable problem size significantly exceeds the potential of current quantum machines.

Here, machines for QA-based approaches are certainly more scalable with respect to the number of qubits that can be applied. For example, the D-Wave Advantage [24] (which is suitable for QA-based approaches) can handle up to 5000 qubits, while gate-based approaches rely on machines (e.g., by IBM or Google) that can handle a bit more than 100 qubits (with very impressive and ambitious roadmaps toward thousands of qubits though).[1] Using, e.g., an one-hot encoding as proposed in [29], this allows QA-based approaches the direct detection of a discrete number k of communities, while, to the best of our knowledge, all gate-based approaches proposed thus far are limited to detecting $k = 2$ communities only. Obviously, this makes comparing the current potential of QA-based and gate-based approaches impossible for identifying an arbitrary number of k communities in a given graph.

Moreover, existing QA-based solutions such as proposed in [29] and [30] have not been evaluated yet on recent hard- and software. Furthermore, Negre et al. (2020) [29] evaluated the algorithm for various real-world networks against a classical annealer, but no time metric is stated which indicates how long it took the algorithms to yield the given results.

Overall, this leaves a situation with multiple approaches utilizing quantum technologies to tackle the community detection problem. However, there is no comprehensive comparison between the, respectively, proposed quantum community detection solutions available yet.

This paper aims to fill the gap by, *first*, contributing two novel algorithms for gate-based devices (for $k \geq 2$ communities), called GB1 and GB2 in the following. *Second*, reimplementing the QA-approaches reviewed above, which are subsequently called QA1 and QA2, utilizing current hard- and software. Finally (*third*), a structured comparative evaluation with standardized real-world and artificially generated benchmark graphs. The following discusses the research questions that guided our research. The utilized algorithms and according differences to previous work are summarized in Table 1.

Research Questions: The overall objective of this paper is to evaluate the current potential of quantum computing for the k-community detection problem. Therefore, the two main prevailing quantum computing architectures are evaluated by comparing their performance and the performance of a classical heuristic algorithm for a given set of test graphs. To enable such a comparison, the creation of novel hybrid algorithms, which are built to run on current gate-based quantum computers, is necessary. This leads to the first research question:

- RQ1: How can possible gate-based hybrid algorithms for k-community detection be constructed?

[1] Note that the number of qubits does not necessarily reflect the power of the quantum device. Other metrics like the connectivity of the qubits or the noise level have to be considered too. To our knowledge, there exists currently no single metric for the overall quality of quantum computers.

Table 1 Hybrid algorithms utilized within this paper

Alg.	Difference to previous work
GB1	The classical framework is conceptually based on [30], but GB1 -utilizes QLS [33] instead of quantum annealing for bipartition to enable k-community detection on gate-based devices -utilizes modularity [8] rather than energy density as the objective function, and -does not perform initial random bipartition of the graph
GB2	QLS [33] is recursively applied for search space reduction.
QA1	Utilizes D-Wave's hybrid solver for discrete quadratic models [26]
QA2	Is a re-implementation of the solution proposed in [30] on current hard- and software

The newly developed algorithms as well as the quantum annealing and classical method are evaluated on a test graph. Furthermore, the impact of the quantum parts within the new gate-based algorithms is assessed via a substitution and subsequent evaluation with simulated annealing methods. Therefore, the following two research questions are answered:

- RQ2: How do the new hybrid gate-based algorithms, the quantum annealing based methods, and the classical benchmark heuristic perform in k-community detection?
- RQ3: What is the performance impact of quantum processing within the gate-based hybrid algorithms?

Due to current hardware limitations [33], the evaluation considers only on a small test graph. However, quantum annealing devices provide a higher number of qubits for combinatorial optimization problems and have already been utilized for a direct encoding of the k-community detection problem [29]. Therefore, we further evaluate the performance of the quantum annealing methods on more advanced larger test graphs, leading to the fourth research question.

- RQ4: How do current quantum annealing based hybrid methods perform on artificially generated benchmark graphs compared to a classical heuristic algorithm?

The new developed hybrid algorithms, together with an implementation of the other algorithms used throughout this paper, are publicly available on Github [17] for reproducibility of the obtained results.

Structure of the paper The remainder of this paper is structured as follows. In Sect. 2, background information concerning the community detection problem and optimization on NISQ devices is given. This also includes a short description of the implemented algorithms which have been proposed in previous work. Section 3 presents the new hybrid algorithms for the k-community detection problem, which utilize gate-based quantum computers, providing an answer to the RQ1 (design question). In Sect. 4, the evaluation methodology, the obtained results of the experiments, and the critical discussion of the obtained results are stated. Therefore, RQ2, RQ3 and RQ4 (knowledge questions) are answered within this section. Finally, the conclusion can be found in Sect. 5. In addition, more details concerning the methodology are given in the appendix.

2 Background

This section provides introductory information concerning the problem of community detection and appropriate mathematical descriptions as well as the optimization approaches on current NISQ devices. Within the later, the focus lies on existent quantum or hybrid algorithms dealing with community detection.

2.1 Community detection problem

The most commonly used measure for the quality of a detected community is the so-called modularity as it has been defined by Newman and Girvan [1,8]. Although the modularity must not be considered as generalizable to all random graphs, the mathematical formulation fits the requirements of quantum computation. Many optimization problems in quantum computing are stated in Ising form [35]. This mathematical abstraction represents the energy of n discrete spin variables $s_i \in \{-1, 1\}$, $1 \leq i \leq n$ with an according local field h_i and interactions J_{ij} between different spins. With this approach, the energy of a configuration is denoted by the Hamiltonian function [35]:

$$H(s) = \sum_{i,j} J_{ij} s_i s_j + \sum_i h_i s_i, \qquad s_i \in \{-1, 1\} \tag{1}$$

An equivalent formulation is the Quadratic Unconstrained Binary Optimization (QUBO) form. With the appropriate variable substitution, one obtains [35]:

$$H(x) = \sum_{i<j} Q_{i,j} x_i x_j + \sum_i Q_{ii} x_i, \qquad x \in \{0, 1\} \tag{2}$$

Using one of these forms apparently constraints the problem to a binary form. For each node of a graph $G = (V, E)$ representing one variable (s_i or x_i), the modularity of at most two communities is therefore given by [1]:

$$M = \frac{1}{2m} \sum_{i,j} B_{ij} \frac{s_i s_j + 1}{2} \tag{3}$$

with m being the sum over all edge weights w_{ij} of the graph and the modularity matrix elements $B_{ij} = A_{ij} - \frac{g_i g_j}{2m}$. In this context, the node degree is $g_i = \sum_j A_{ij}$ where A_{ij} represents the according adjacency matrix elements. Similarly, the modularity for a discrete number of communities is given by [8]:

$$M = \frac{1}{2m} \sum_{i,j} B_{ij} \delta(c_i, c_i) \tag{4}$$

2.2 Optimization on NISQ devices

The most prominent quantum approaches designed for NISQ devices are the Quantum Approximate Optimization Algorithm (QAOA) and Quantum Annealing (QA), both of which are inspired by the adiabatic theorem [35]. In QA, the system starts in the ground state of the initial Hamiltonian H_I and evolves according to [35]:

$$H(t) = \frac{t}{T} H_C + \left(1 - \frac{t}{T}\right) H_I, \quad t \in \{0, T\} \tag{5}$$

to the final state, which is denoted by H_C in Ising form (Eq. 1). QAOA applies the logic of QA to gate-based quantum computers. Here, the time evolution operator of the system is represented by an alternating application of quantum operators, which are constructed by either H_M or H_C and according parameters β, γ. The resulting trial states $\Psi(\beta, \gamma)$ for different parameter settings are measured, and the parameters are optimized classically to produce the ground state of H_C [35]:

$$\beta_*, \gamma_* = argmin_{\beta,\gamma} \langle \Psi(\beta, \gamma)|H_C|\Psi(\beta, \gamma)\rangle \tag{6}$$

An existent hybrid quantum-classical algorithm for community detection which implements QA or QAOA for sub-problem optimization is the Quantum Local Search (QLS) algorithm [33], which has also been implemented within a multi-level approach [35]. The algorithm starts with a random solution and iteratively populates subsets, which are optimized by QA or QAOA. The sub-problem is formulated by fixing the community assignment for all nodes which are not in the subset and encoding them as boundary conditions. The convergence criterion is defined by three iterations without any further improvement in modularity.

For quantum annealing devices only, the concept of logical super-nodes has been introduced, which allows to handle the k-community detection problem on such devices [29]. The higher number of available qubits enables the use of one-hot encoding. This approach has been tested with the D-Wave 2000Q and the qbsolv software. However, the recently launched D-Wave Advantage and the hybrid solver for discrete quadratic models (refer to the appendix), which builds on the super-node logic, have not been tested for community detection so far. These technologies have been used in the first quantum annealing approach (QA1) in the experiments stated in Sect. 4.

An alternative approach which utilizes quantum annealing has been suggested by Reittu et al. (2020) [30]. The algorithm is based on the Szemeredi’s Regularity Lemma, which is defined for large bipartite graphs. After an initial bipartition of the given graph, iteratively the strongest communities (based on the connectivity) are detected and removed from the graph. In contrast to other algorithms, the underlying metric for optimization is not the modularity but rather the difference in energy density compared to a mean. The D-Wave 2000Q and qbsolv have been utilized for the experiments. An implementation of this approach represents our second quantum annealing method (QA2).

3 Methods for gate-based k-community detection

To the best of our knowledge, there is currently a lack of hybrid algorithms for the k-community detection problem, which can be run on gate-based devices. For the purpose of evaluating the potential of currently available quantum architectures, novel algorithms have been created, which are inspired by previous work [29,30,33].

3.1 Density-based community detection algorithm

The first gate-based algorithm (GB1) works similarly to the one proposed by Reittu et al. (2020) [30] as it iteratively finds the strongest communities within a given graph. However, GB1 is designed for gate-based devices rather than quantum annealing machines and does not bipartite the graph randomly. Therefore, the Szemeredi's Regularity Lemma which addresses large bipartite graphs cannot be taken as a mathematical foundation for this algorithm. However, for smaller graphs a random bipartition affects the community structure more likely and more severe. For this reason, which has also been tested experimentally, we decided to omit the initial bipartition of the graph. This experiments have been conducted for design purposes and are therefore not stated in the paper.

Due to the higher number of available qubits in quantum annealing devices, the algorithm proposed by Reittu et al. (2020) [30] can directly solve subproblems (the identification of just two communities for a given set of nodes in the graph) via quantum annealing. However, for gate-model machines the use of a hybrid algorithm which can handle a larger number of vertices for these subproblems is necessary. We implemented QLS [33] for this task. Furthermore, to assure that all proposed algorithms optimize the same objective function, the modularity (Eq. 4) has been taken for this purpose rather than an energy density deviation from the mean as it is used in the Szemeredi's Regularity Lemma. The algorithm is outlined in Algorithm 1.

Algorithm 1 Density-based community detection algorithm (GB1)

```
function GB1(G)
  $n = |G|$
  $k = int(\sqrt{n}) + 1$
  for je in 1,...,k:$\rightarrow$ tries to find communities in k rounds
    if $(je > 1)$: remove community $V^{'}$ from G
    if $(|G| = 0)$: stop
    $m1 = m2 = 0; \quad V = V^{'} = G$
    while $(m1 \geq m2)$ :$\rightarrow$ stop at max. modularity per node
      $m2 = m1; \quad V^{'} = V$
      prepare modularity matrix for current iteration: $c = mod\_matrix(V')$
      QLS(c) $\rightarrow$use QLS to find new community V with modularity m
      if $(|V| = 0)$: break
      $n = |V|; \quad m1 = \frac{m}{n}$
    store $V^{'}$ as community
  return communities
```

It is assumed that the maximum number of communities does not exceed $\sqrt{n}$ with n being the number of nodes in the graph [36]. The loop searches for the strongest community, with the modularity per node as the underlying metric. This criterion is not based on mathematical analytical reasoning, but rather on heuristics. After the strongest community has been detected, it is stored and the according nodes are deleted from the original graph before the procedure repeats for the remaining nodes. The algorithm stops if the graph does not contain vertices anymore. An alternative breaking scenario occurs in case there is no initial improvement possible within the inner loop (i.e., $m < 0$). It is assumed that the algorithm works best for graphs with heterogeneous community structure, because of the logic of iteratively detecting communities based on their connectivity.

3.2 Recursive-QLS community detection algorithm

The second gate-based algorithm (GB2) represents a recursive application of QLS. Here, the original system is separated into 2^x communities using the recursive_QLS function as it is outlined in Algorithm 2. In each recursion, the function adds an attribute to all individual nodes according to the binary community assignment. The sub-graphs detected in that way serve as the graph parameter in the subsequent recursion. After a defined number of recursions has been executed, the given attributes of a node encode its final community. In our approach, we define this upper boundary as the next higher integer after $\log_2(k)$, where again k is given by $\sqrt{n}$.

Using GB2, the maximum modularity can then be found by a brute-force search in the space defined by the power set with the detected communities as its elements. Alternatively, one could look for the highest modularity in the set of community constellations of the respective recursions by iteratively skipping the last node attribute. Which alternative is appropriate depends on the existing knowledge concerning the topology and community structure of the graph. Although the scalability of the proposed algorithms is not within the scope of this paper, GB2 could be applied to larger graphs by considering the initially detected communities as nodes in a subsequent optimization iteration similarly like it happens within the Louvain algorithm [3]. Thereby, a considerable part of the limitations in the search space, which are induced by the recursive GB2, could be compensated.

However, the mentioned search space limitations suggest, that GB2 works best for a graph with 2^x communities. Furthermore, it is expected that GB2 is faster than GB1, because the number of nodes which have to be optimized in a binary manner reduces faster with each recursion of GB2 than it does for the iterations in GB1.

Algorithm 2 Recursive QLS

```
function recursive_QLS(G,k,iter=0)
    if ($|G| = 0$): stop
    QLS(G) → separate given graph into 2 communities
    store values of QLS-result (0 or 1 for each node) as node attribute
    $G1$ = nodes with attribute 1
    $G2$ = nodes with attribute 0
    $iter = iter + 1$
    if ($iter \geq k$): stop
    recursive_QLS($G1, k, iter$)
    recursive_QLS($G2, k, iter$)
```

4 Experiments and results

The overall objective of the paper is to evaluate the current potential of gate-model and quantum annealing based devices for the k-community detection problem. The introduction of the novel hybrid algorithms for gate-based devices (Sect. 3) enables an initial comparative evaluation of the different architectures.

The conducted experiments are twofold. The first part considers the performance of all algorithms that have been described above on a real social network, referring to research question RQ2 and RQ3. As the hybrid solver for discrete quadratic models [26] is considered the most suitable for scaled up problems, the algorithm is further tested and compared with a classical state-of-the-art algorithm on four artificial test graphs in the second part. Therefore, research question RQ4 is answered within this scope. The subsequent chapters deal with the underlying methodology of the experiments, the obtained results and the discussion of those results.

4.1 Methodology

General procedure. In order to assess the performance of the algorithms described above, they have been applied to standardized graphs. The metrics of interest are

- Time-to-solution: How long does it take the algorithm to find the solution?
- Quality: What is the modularity of the obtained solution as it has been defined in Sect. 2.1. (Eq. 4)?

Every experiment has been conducted five times in order to be able to state an average, best and worst result for each constellation. It has been criticized in [12] that in almost any paper, where a new method for community detection is introduced, the testing consists in applying the method to a small set of simple benchmark graphs. However, within the scope of this paper we have to follow this tradition by evaluating the algorithms on one prominent real-world dataset or several artificial benchmark graphs, respectively. Further research on the scalability and specificity of the proposed algorithms may be the subject of future work. The general framework is implemented in Python 3.8 with NetworkX for network analysis operations. For statistical testing of the obtained results, the Mann–Whitney U test [2] has been used, which is equivalent [23] to the Wilcoxon ranksum test. This test allows comparison between two solution sets without the assumption of normal distribution. The significance level has been chosen to be 95% ($\alpha = 0,05$). The chosen classical state-of-the-art algorithm for community detection for benchmarking is the Louvain algorithm [3,12].

First part. Every method described above (QA1, QA2, GB1, GB2, Louvain) has been tested on the Zachary Karate Club Graph [37], which consists of 34 nodes and 78 edges. This graph has been extensively used to benchmark graph algorithms and is considered to be by far the most investigated system [12]. The various methods utilize different hardware and software. However, our framework is modular and can easily be extended to utilize other quantum computing architectures as they become available. For details concerning the used technologies, we kindly refer the interested reader to the appendix.

Within the first quantum annealing approach (QA1), the problem is solved by the D-Wave Hybrid Solver Service [26] with the D-Wave Advantage quantum annealer [24] as the underlying hardware. This hybrid solver allows to directly address discrete quadratic problems. Due to a high degree of automation, a parameter tuning like it has been conducted in [29] was not possible. The implementation of QA2 also runs on the D-Wave Advantage quantum annealer with the parameters set to the default values. Automatically minor embedding has been used, which is re-calculated each time a sampling method is called. The simulated annealing methods make use of the dwave-neal software tool. Therefore, the problem runs on servers of Amazon Web Services. This shall enhance the comparability of the obtained results as currently running quantum algorithms represent a form of online computation.

Concerning the gate-based algorithms, these utilize the Qiskit Aqua library, which provides an implementation of QAOA with the MinimumEigenOptimizer as a wrapper. Another implementation of QAOA is publicly available [17], but has not been used in the experiments. Communication with the hardware happened via the Qiskit IBM Quantum Provider. The problems have been sent to the IBM Melbourne quantum computer (15 qubits) with a QLS subset size of 12. The convergence criterion in QLS has been set to 3 iterations. To estimate the impact of the subset size on the number of required iterations, tests have been conducted with a subset size of 16 using simulated annealing. Hereby, one iteration is defined as one sub-problem optimization.

It has to be mentioned, that different definitions of the time-to-solution have been used. In QA1 and the Louvain algorithm, the overall time required to obtain a solution has been regarded, which is more meaningful for practical applications. Within QA2, GB1 and GB2, the time-to-solution is defined as the sum of all quantum processing (or simulated annealing) parts within the hybrid algorithm. Therefore, the queueing time is not considered in the metric, which makes it more representative. Furthermore, the biases of local computation differences are reduced. Additional information concerning timing issues of the utilized quantum systems are stated in the appendix. This mixed evaluation is justified by the overall objective of the paper, which is to assess the current potential of quantum technologies for the k-community detection problem in general, thus allowing to compare the different state-of-the-art of quantum annealing and gate-based methods.

Second Part. The second part extends the evaluation of the QA1. For this purpose, 12 test graphs of different size and density have been generated. These are the planted l-partition graph [9] and the LFR-benchmark graph [20], which are considered prominent artificial graphs for testing community detection algorithms [1,12]. The planted l-partition graph shows a homogeneous community structure, whereas the LFR-benchmark graph is heterogeneous in community size and connectivity [12]. Both types of graphs have been generated with 100, 246, 496, 747, 1000 and 1500 nodes, with parameters that ensure a rather well-defined community structure and stable degree (for further details please consult to the appendix).

Table 2 Average, best and worst performance values of hybrid algorithms and the Louvain algorithm for the Zachary Karate Club Graph; results for simulated annealing are stated in brackets

Algorithm	Modularity	Time (ms)
QA1	[0,4198; 0,4198; 0,4198]	[5001; 4984; 5030]
QA2	[0,1462; 0,1824; 0,0756]	[425; 453; 318]
	([0,2142; 0,2530; 0,1829])	([180; 250; 127])
GB1	[0,3900; 0,4015; 0,3740]	[1078544; 995670; 917493]
	([0,3940; 0,4172; 0,3734])	([608; 876; 605])
GB2	[0,4061; 0,4188; 0,3969]	[622580; 659579; 570063]
	([0,4059; 0,4198; 0,3941])	([333; 274; 310])
Louvain	[0,4096; 0,4188; 0,3969]	[12; 14; 15]

Table 3 Average, lowest and highest number of iterations for different subset sizes

Algorithm	#for subset size=12	#for subset size=16	Stat. Sign.
GB1	[80,8; 62; 118]	[62,8; 43; 75]	yes
GB2	[46,6; 43; 51]	[44,4; 44; 46]	no

Table 4 Statistical significance of the compared algorithms regarding the obtained modularities

Algorithm	QA1	QA2	GB1	GB2	Louvain
QA1	–	Yes	Yes	Yes	Yes
QA2	Yes	–	Yes	Yes	Yes
GB1	Yes	Yes	–	Yes	No
GB2	Yes	Yes	Yes	–	No
Louvain	Yes	Yes	No	No	–

4.2 Results

This section states the results of the experiments, which are going to be discussed in the subsequent chapter. All raw result data are publicly available [17]. The experimental results concerning the first part are stated in Table 2 for the average, best and worst performance values out of the 5 runs, based on the modularity. All time values are in milliseconds. The obtained results for simulated annealing are stated in brackets. Furthermore, the results concerning the impact of the subset size in QLS are stated in Table 3.

Table 4 provides information concerning the statistical significance of the obtained modularity values yielded by the compared algorithms. All time values are considered to be significant. The modularities obtained by utilizing the IBMQ Melbourne or simulated annealing for sub-problem optimization show no significant difference for GB1 and GB2.

The experimental results concerning the second part are stated in Table 5 for the average, best and worst performance values, based on the modularity. Time values

Table 5 Average, best and worst performance values of the QA1 and Louvain algorithm for different artificial graphs

Artif. graph	QA1 modularity	Louvain modularity	Stat. Sign.
PP100	[0,3341; 0,3341; 0,3341]	[0,3341; 0,3341; 0,3341]	No
LFR100	[0,143; 0,1471; 0,1410]	[0,1392; 0,1429; 0,1358]	No
PP246	[0,3270; 0,3270; 0,3270]	[0,3270; 0,3270; 0,3270]	No
LFR246	[0,2037; 0,2037; 0,2037]	[0,1874; 0,1918; 0,1791]	Yes
PP496	[0,2772; 0,2772; 0,2772]	[0,2818; 0,2824; 0,2807]	Yes
LFR496	[0,4017; 0,4017; 0,4017]	[0,4017; 0,4017; 0,4017]	No
PP747	[0,2700; 0,2700; 0,2700]	[0,2711; 0,2733; 0,2693]	No
LFR747	[0,5372; 0,5372; 0,5372]	[0,5372; 0,5372; 0,5372]	No
PP1000	[0,1054; 0,1095; 0,0990]	[0,1818; 0,1867; 0,1791]	Yes
LFR1000	[0,3536; 0,3588; 0,3449]	[0,3601; 0,3605; 0,3591]	Yes
PP1500	[0,1827; 0,1902; 0,1774]	[0,1737; 0,1772; 0,1699]	Yes
LFR1500	[0,1768; 0,1772; 0,1763]	[0,1623; 0,1646; 0,1584]	Yes
Artif. graph	**QA1 Time (ms)**	**Louvain Time (ms)**	**Stat. Sign.**
PP100	[5361; 5347; 5390]	[38; 30; 60]	Yes
LFR100	[5353; 5342; 5354]	[75; 56; 91]	Yes
PP246	[30676; 30568; 31039]	[129; 112; 141]	Yes
LFR246	[30739; 30581; 30933]	[226; 190; 350]	Yes
PP496	[174667; 174653; 174693]	[520; 397; 715]	Yes
LFR496	[176167; 176125; 176195]	[287; 238; 391]	Yes
PP747	[274837; 274727; 274886]	[980; 568; 1201]	Yes
LFR747	[275009; 274854; 275157]	[424; 400; 470]	Yes
PP1000	[354189; 353814; 354558]	[1493; 1288; 2037]	Yes
LFR1000	[354509; 354124; 354771]	[1994; 1789; 2321]	Yes
PP1500	[477893; 477243; 478775]	[3293; 3989; 3045]	Yes
LFR1500	[477855; 477196; 478281]	[2231; 2489; 2017]	Yes

are given in milliseconds. Table 5 additionally states, whether the difference in the obtained modularity values can be considered as statistically significant.

4.3 Discussion of results

The discussion is structured according to the two parts of the experimental setup: the first one considering the performance on the Zachary Karate Club Graph only (RQ2, RQ3), and the second one extending the experiments for QA1 to artificially generated test graphs (RQ4).

To our knowledge, the best modularity value for the Zachary Karate Club Graph is 0,4198 [3,6], resulting in 4 communities. The subsequent interpretation of the results requires to consider the following aspects:

- Varying certain elements of the hybrid algorithms (e.g., different QAOA implementation, different subset size and convergence criterion in QLS, parameter tuning in QA2) may significantly alter the performance of the algorithms, although they have been set consciously.
- The characteristics of the Zachary Karate Club Graph, with its 4 communities, fit the recursive nature of GB2. However, QA2 struggles with such small graphs due to the applicability of the Szemeredi's Regularity Lemma and the initial random bipartition of the network.
- The Louvain algorithm is explicitly designed to solve the community detection problem especially for larger networks. GB1, GB2, QA1 and QA2 just need minor adoptions to the underlying optimization function to be more generally applicable.

Answering RQ2. The QA1 is able to reach the highest known modularity in all runs, but needs considerable time to do so. Due to the arguments mentioned above, the QA2 performs rather poor on the small Zachary Karate Club Graph. More interestingly, GB1 and GB2 both show no significant difference in the obtained modularities compared to the state-of-the-art classical Louvain algorithm. Although GB2 performs significantly slightly better than GB1 in this particular experiment, the later is considered the more versatile algorithm.

The values for the time-to-solution appear to vary notably between the utilized algorithms. A decrease in the time-to-solution for GB1 and GB2 could happen via a lower number of iterations per run. This in turn is indicated by the results to be possible with larger subset sizes. However, the downturn of larger sub-problems may be longer times for the solution per iteration. Which effect dominates in practice may be the subject of further studies. How different subset sizes may affect not only the number of iterations but also the obtained modularity and time-to-solution could be roughly extracted from the data provided on Github [17], but does not lie within the focus of this paper.

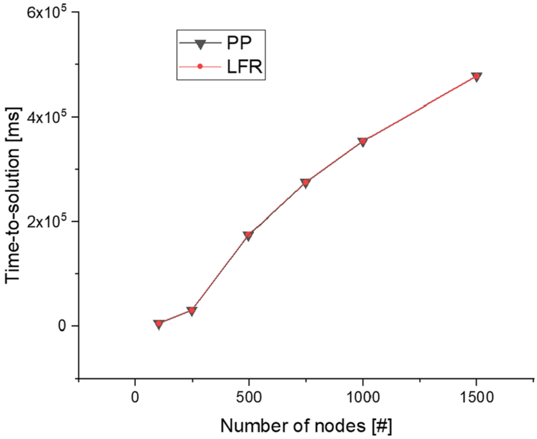
Lastly, it shall be mentioned that due to practical run-time reasons, the maximum number of iterations in the inner loop of GB1 has been set to 5. This number has sometimes been reached in our experiments, indicating that with more computation time even better modularity values may be possible.

Answering RQ3. Substituting QAOA within GB1 and GB2 with simulated annealing shows no significant differences in the obtained modularity values, which indicates a rather high solution quality of the IBMQ Melbourne concerning small problems. The values for the time-to-solution on the other hand vary notably with the use of simulated annealing.

For this reason, the proposed hybrid classical-quantum algorithms are considered to yield practical value only after the sub-problem optimization by QAOA exceeds simulated annealing methods, which heavily depends on future advances on the physical hardware level.

Answering RQ4. Concerning the obtained modularity values, it can be observed that for smaller graphs (e.g., PP100, PP246, LFR496) the QA1 and Louvain Algorithm (LA) yield equal values. This indicates that the global optimum may has been found,

Fig. 1 Time-to-solution against number of nodes for the QA1



or both get easily trapped in the same local minimum. For graphs of up to 747 nodes, the QA1 yields equal results for every run which are better or worse than the ones of the LA and are probably caused by same partitions of the graph. The obtained results for larger graphs show a higher variance for both algorithms which may be caused by the according larger search spaces. This may also be the reason for the statistical differences when it comes to larger problem instances. What could also be noticed is a trend for better performance of the QA1 for more heterogeneous graphs (LFR-graphs). This may be caused by a potentially larger spectral gap between the different community assignments which in turn would be favorable for quantum annealing. Concerning the obtained time-to-solution values, the QA1 scales linearly or even slightly better as it is shown in Fig. 1, whereas the LA appears to run in time O(n*logn) [19]. Figure 1 shows that there is almost no difference in the obtained values for PP and LFR graphs. This may be caused by the hybrid solver framework of D-Wave which unfortunately does not provide transparent analysis of the conducted steps. Looking at the time values in Table 5 clearly shows the longer, but more robust, running times of QA1.

Overall, it was not possible to identify an remarkable increase in solution quality or time-to-solution with the QA1 compared to a state-of-the-art heuristic classical solver. However, as [25] claims accelerated time-to-solutions for very large problems, it would be interesting to extend the evaluation to graphs of bigger sizes.

But not only the graph size and density shall be considered. The obtained results show that there are significant performance differences which arise only due to the community structure of the graph, which is defined by its homogeneity, average degree, ratio of intra- and inter-community connectivity, etc. Therefore, as criticized in [12], the specificity of the algorithms for given problems has to be examined without restrictions to size and density.

Finally, it should be mentioned that both the D-Wave Advantage and the hybrid solver for DQM where utilized in their first version, suggesting space for near-term efficiency improvements.

4.3.1 Threats to validity

Although already partially mentioned in the previous discussions, this section provides a structured review of the threads to validity following the scheme of [31].

Construct validity. To assess the current potential of quantum computing for the k-community detection problem, existing and novel algorithms have been utilized. However, varying parameters in the hybrid algorithms or optimizing pre- and post-processing of each quantum query [21] could affect the obtained results. These effects are not within the scope of this work, because hybrid applications for practical usage require automatized parameter settings. Therefore, the automatic parameter setting by the vendors has been used or a conscious setting of the given parameters has been conducted (e.g., a subset size of 12 to account for ancillary qubits [21]).

Internal validity. Concerning the obtained modularity results of the substitution with simulated annealing, we regard the quality of the IBMQ Melbourne as the prevailing cause. However, there is still the possibility of leveling of poor results by the hybrid algorithm. As the research question concerns only the affect within the hybrid algorithm, the quality of the QAOA results has not been further investigated.

External validity. The proposed hybrid algorithms should be considered just as prototypes of possible quantum computing approaches for tackling the k-community detection problem. The mentioned specificity and dependence on the given problem, the parameter setting, as well as the underlying quantum parts make generalizing a difficult endeavor. In this sense, QAOA and quantum annealing are still to be considered in an early stage where their characteristics are being figured out currently (e.g., [38], [22]) and improvements are made on a regular basis (e.g., [39], [7]). For this reasons, we expect the algorithms to show similar performance when applied to similar problem instances as those examined in this work. However, we cannot state further generalizations as a sound sensitivity analysis is not within the scope of this paper, but may be a subject of future work. Furthermore, in the first part of this evaluation, no statements concerning the scaling of the proposed algorithms are made. Accounting for the criticism in [12], we stated the graph features carefully and varied certain parameters only while keeping others stable to ensure a higher degree of causality. Specific details are stated in previous sections and in the appendix.

Reliability. Although dealing with heuristic algorithms, which are stochastic in nature, we specified the used technologies and methodology in detail and provide all code and raw data on Github [17] for reproducibility. In order to be able to state statistically significant results, the Wilcoxon ranksum test has been applied.

5 Conclusion

In this paper, we have evaluated the potential of current NISQ devices for the k-community detection problem. Therefore, we have proposed novel hybrid algo-

Table 6 Results at a glance

RQ	Answer
RQ1	GB1: based on local graph density, iteratively finds strongest community
	GB2: recursive application of QLS-algorithm, best for 2^x communities
RQ2	Modularity: QA performs best, GB1 & GB2 similar to classical solver
	Time-to-solution: significantly higher for all quantum methods
RQ3	Modularity: no significant difference
	Time-to-solution: significantly less with simulated annealing
RQ4	Modularity: similar; Time-to-solution: significantly higher for QA

rithms for gate-model devices and compared their performance to existing quantum annealing-based approaches and a state-of-the-art classical heuristic solver. We ran our experiments on the 15 qubit IBMQ Melbourne and the D-Wave Advantage devices. Furthermore, the quantum processing parts have been substituted with simulated annealing to obtain information concerning the impact of quantum computing within the hybrid algorithms. The obtained answers to the given research questions are summarized in Table 6.

Considering the modularity obtained with the various methods on the Zachary Karate Club Graph (RQ2), the QA1 which utilizes the D-Wave hybrid solver for discrete quadratic models performs best. GB1 and GB2 show no significant difference to the classical benchmark method (Louvain algorithm). The substitution with simulated annealing (RQ3) does not yield any significant difference in modularity, whereas the time-to-solution is notably decreased.

Utilizing the hybrid solver for discrete quadratic models provided by D-Wave, we were able to obtain modularity values similar to those of the Louvain algorithm, but got much longer time-to-solution values (RQ4). Furthermore, the results show that not only the graph size and density should be considered in the evaluation due to the significant impact of other factors which are related to the underlying community structure. These may be highly specific to each utilized algorithm.

Overall, quantum computing for the k-community detection problem has been figured out to yield good modularity values for small graphs, but still the computation times are too high to be of practical relevance. Given the current and near-term hardware limitations, we consider versatile and modularizable hybrid quantum classical algorithms, like the ones outlined, to be crucial for proceeding toward practical usage of NISQ devices, and therefore, for justification of quantum computing in general. As the potential of available NISQ devices increases, we are optimistic that similar methods can lead to a broad utilization of quantum computing to a variety of real-world applications.

Future Work: We are currently experimenting with quantum algorithms other than QAOA, such as the Grover Adaptive Search [15], in our proposed hybrid algorithms. Furthermore, we are working on a concept, how to model the use of these algorithms according to Model Driven Software Engineering principles [5]. In this sense, we will investigate how to reuse existing algorithms for a particular problem, but also how to model the hybrid workflows of solving combinatorial optimization problems on NISQ devices. The scalability and specificity of the proposed algorithms may be an additional subject of further studies.

Funding Open access funding provided by Johannes Kepler University Linz.

Availability of data and material All raw data are publicly available on Github [17].

Code availability All source code is publicly available on Github [17].

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Funding This work was partially funded by the Austrian Federal Ministry for Digital and Economic Affairs and the National Foundation for Research, Technology and Development. Furthermore, this work has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 101001318) and was partially been supported by the LIT Secure and Correct Systems Lab funded by the State of Upper Austria as well as by the BMK, BMDW, and the State of Upper Austria in the frame of the COMET program (managed by the FFG).

6 Appendix

This section provides more details concerning the hardware and software used throughout the experiments as well as timing information concerning the utilized quantum systems. Furthermore, information about the generated benchmark graphs is given. The reason for choosing 5 times as the number of repetitions for each constellation is twofold. First, one run on the IBM hardware needs an effective time of approx. one day. Second, this number allowed the evaluation utilizing D-Wave technology without the necessity to acquire extensive computation time.

D-Wave systems:

For QA1, the Hybrid Solver Service provided by D-Wave has been utilized, which allows to directly address discrete quadratic models (DQM) [26]. The hybrid solvers implement state-of-the-art classical algorithms together with intelligent allocation of the quantum processing unit. This high degree of automation comes with a lack of adjustable parameters and transparency. The hybrid DQM solver version 1 can read discrete problems with up to a maximum of 2 billion total input weights [26]. Like with the other hybrid solvers offered by D-Wave, queries can be sent to the Advantage quantum processing unit. This quantum device has been launched in September 2020 and contains 5000 qubits, which is about 2.5 times more than in its predecessor, the D-

Wave 2000Q. Another important difference lies in the connectivity enhancing Pegasus typology of the device, which comes with 15 couplers per qubit instead of 6 [24]. The Advantage-System 1.1 has been used throughout this project. The time per instance sent to the D-Wave Advantage within QA2 is defined as the QPU access time plus an post-processing overhead time, where the QPU access time consists of the following [35].

$$\begin{aligned}qpu\_access\_time &= qpu\_programming\_time\\ &+N*(annealing\_time+delay\_time+readout\_time)\end{aligned}$$

The first is a one-time initialization step to program the QPU once per call. N denotes the number of samples which linearly impacts the QPU sampling time. The adjustable parameters for the experiments have been set to the default values according to D-Wave Hybrid 0.6.1 Documentation ($N = 100$, $annealing\_time = 20\mu s$). Adjustment of these parameters may lead to significant changes in the overall performance of the QA2 but has not been studied within this project.

IBM systems:

The Qiskit library is an open-source framework which allows to work with IBM quantum computers[2]. For the experiments, Qiskit 0.23.2 has been used. Within Qiskit, the Qiskit Aqua software tool represents the element of the library for building algorithms and applications such as QAOA. The IBM Melbourne quantum computer consists of 15 qubits and reaches a quantum volume of 8. It therefore cannot be regarded a state-of-the-art quantum computer (e.g., IBM claimed to have reached a quantum volume of 64 with a 27 qubits system in August 2020[3]). However, it represents the most powerful device which is publicly available. Version 2.3.4 has been used throughout this project. The time-to-solution of GB1 and GB2 heavily depends on the parameters of the QLS algorithm. Varying the size of the subsets and the convergence criterion significantly impacts the number of iterations and therefore should have an effect on the time-to-solution metric. However, smaller subsets may also lead to faster results per iteration on the quantum computer. Furthermore, the implementation using the MinimumEigenOptimizer creates multiple equal jobs for one optimization problem with 1024 shots each. An alternative implementation like the one provided on Github [17] may be more efficient in that sense. A further possibility of improving the QAOA efficiency is via tensor-network based approaches which allow classical training of certain QAOA instances [34] . In this project, only the running time of the instances sent to the IBM device has been considered. The queuing, validation, and creation times have been omitted.

Artificially generated benchmark graphs:

The planted l-partition model partitions a graph with $n = g * l$ nodes into l groups consisting of g vertices each. The nodes within one group are linked with probability p_{in}, whereas the nodes of different groups are linked with probability p_{out}. Therefore, the average degree of a node can be calculated via $\langle k\rangle = p_{in}*(g-1)+p_{out}*g*(l-1) =$

[2] https://qiskit.org/.24.06.2021

[3] https://newsroom.ibm.com/2020-08-20-IBM-Delivers-Its-Highest-Quantum-Volume-to-Date-Expanding-the-Computational-Power-of-its-IBM-Cloud-Accessible-Quantum-Computers.24.06.2021

$z_{in} + z_{out}$ with $z_{in} = p_{in}(g-1)$ and $z_{out} = p_{out} * g * (l-1)$. If $p_{in} > p_{out}$, the intra-group edge density exceeds the inter-group edge density and the graph shows a community structure [12]. For the graphs in this paper the following parameters have been used to ensure a stable ratio of $\frac{p_{in}}{p_{out}}$ and a relatively stable average degree:

- PP100: $l = 5, \quad g = 20, \quad p_{in} = 0.8, \quad p_{out} = 0.16 \rightarrow p_{in} = 5 * p_{out}, \quad z_{in} = 15.2, \quad z_{out} = 12.8, \quad \langle k \rangle = 28$
- PP246: $l = 6, \quad g = 41, \quad p_{in} = 0.3, \quad p_{out} = 0.06 \rightarrow p_{in} = 5 * p_{out}, \quad z_{in} = 12, \quad z_{out} = 12.3, \quad \langle k \rangle = 24.3$
- PP496: $l = 8, \quad g = 62, \quad p_{in} = 0.18, \quad p_{out} = 0.036 \rightarrow p_{in} = 5 * p_{out}, \quad z_{in} = 10.98, \quad z_{out} = 15.62, \quad \langle k \rangle = 26.6$
- PP747: $l = 9, \quad g = 83, \quad p_{in} = 0.125, \quad p_{out} = 0.025 \rightarrow p_{in} = 5 * p_{out}, \quad z_{in} = 10.25, \quad z_{out} = 16.6, \quad \langle k \rangle = 26.85$
- PP1000: $l = 20, \quad g = 50, \quad p_{in} = 0.1, \quad p_{out} = 0.02 \rightarrow p_{in} = 5 * p_{out}, \quad z_{in} = 4.9, \quad z_{out} = 19, \quad \langle k \rangle = 23.9$
- PP1500: $l = 15, \quad g = 100, \quad p_{in} = 0.075, \quad p_{out} = 0.015 \rightarrow p_{in} = 5 * p_{out}, \quad z_{in} = 7.425, \quad z_{out} = 21, \quad \langle k \rangle = 28.425$

The LFR-benchmark graph is more heterogeneous than the planted l-partition graph, assuming that the distributions of degree and community size follow power laws, with exponents τ_1 and τ_2. The mixing parameter μ $(0 \leq \mu \leq 1)$ denotes the fraction of a node's edges which it shares with nodes of other communities [12]. The parameter values for typical networks are reported [20] to be $2 \leq \tau 1 \leq 3$ and $1 \leq \tau 2 \leq 2$. To ensure similar properties compared to the planted l-partition graph, a parameter fit has been conducted. The parameters used for the experiments are:

- LFR100: $\tau 1 = 2.5, \quad \tau 2 = 1.5, \quad \mu = 0.17, \quad \langle k \rangle = 28$
- LFR246: $\tau 1 = 2.5, \quad \tau 2 = 1.5, \quad \mu = 0.17, \quad \langle k \rangle = 24.3$
- LFR496: $\tau 1 = 2.5, \quad \tau 2 = 1.5, \quad \mu = 0.17, \quad \langle k \rangle = 26.6$
- LFR747: $\tau 1 = 2.5, \quad \tau 2 = 1.5, \quad \mu = 0.17, \quad \langle k \rangle = 26.85$
- LFR1000: $\tau 1 = 2.5, \quad \tau 2 = 1.5, \quad \mu = 0.17, \quad \langle k \rangle = 23.9$
- LFR1500: $\tau 1 = 2.5, \quad \tau 2 = 1.5, \quad \mu = 0.17, \quad \langle k \rangle = 28.425$

References

1. Akbar, S., Saritha, S.K.: Towards quantum computing based community detection. Comput. Sci. Rev. (2020). https://doi.org/10.1016/j.cosrev.2020.10031
2. Arcuri, A., Briand, L.: A practical guide for using statistical tests to assess randomized algorithms in software engineering. In: 33rd International Conference on Software Engineering (ICSE) (2011). https://doi.org/10.1145/1985793.1985795
3. Blondel, V.D., Guillaume, J., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. J. Stat. Mech. Theory Experiment. (2008). https://doi.org/10.1088/1742-5468/2008/10/P10008
4. Bork, D., Garmendia, A., Wimmer, M.: Towards a Multi-Objective Modularization Approach for Entity-Relationship Models. ER Forum, Demo and Posters 2020, co-located with 39th International Conference on Conceptual Modeling (ER) (2020)
5. Brambilla, M., Cabot, J., Wimmer, M.: Model-Driven Software Engineering in Practice: Second Edition. Morgan & Claypool Publishers (2017). https://doi.org/10.2200/S00751ED2V01Y201701SWE004
6. Cafieri, S., Hansen, P., Liberti, L.: Improving heuristics for network modularity maximization using an exact algorithm. Discrete Appl. Math. (2014). https://doi.org/10.1016/j.dam.2012.03.030

7. Chen, Y.Q., Chen, Y., Lee, C.K., Zhang, S., Hsieh, C.Y.: Optimizing Quantum Annealing Schedules: From Monte Carlo Tree Search to QuantumZero. arXiv:2004.02836. (2020)
8. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. Phys. Rev. E Stat. Nonlinear Soft Matter Phys. (2004). https://doi.org/10.1103/PhysRevE.70.066111
9. Condon, A., Karp, R.M.: Algorithms for Graph Partitioning on the Planted Partition Model. In: Hochbaum D.S., Jansen K., Rolim J.D.P., Sinclair A. (eds) Randomization, Approximation, and Combinatorial Optimization. Algorithms and Techniques. RANDOM 1999, APPROX 1999. Lecture Notes in Computer Science, vol 1671. Springer, Berlin, Heidelberg (1999). https://doi.org/10.1007/978-3-540-48413-4_23
10. Farhi, E., Goldstone, J., Gutmann, S.: A Quantum Approximate Optimization Algorithm. arXiv:1411.4028. (2014)
11. Ferrara, E., De Meo, P., Catanese, S., Fiumara, G.: Detecting criminal organizations in mobile phone networks. Expert Syst. Appl. (2014). https://doi.org/10.1016/j.eswa.2014.03.024
12. Fortunato, S.: Community detection in graphs. Phys. Rep. (2010). https://doi.org/10.1016/j.physrep.2009.11.002
13. Fortunato, S., Hric, D.: Community detection in networks: A user guide. Phys. Rep. (2016). https://doi.org/10.1016/j.physrep.2016.09.002
14. George, R., Shujaee, K., Kerwat, M., Felfli, Z., Gelenbe, D., Ukuwu, K.: A comparative evaluation of community detection algorithms in social networks. Procedia Comput. Sci. (2020). https://doi.org/10.1016/j.procs.2020.04.124
15. Gilliam, A., Woerner, S., Gonciulea, C.: Grover Adaptive Search for Constrained Polynomial Binary Optimization. arXiv:1912.04088v2. (2019)
16. Girvan, M., Newman, M.: Community structure in social and biological networks. Proc Natl Acad Sci. 99(12) (2001). https://dx.doi.org/10.1073%2Fpnas.122653799
17. Github. Quantum Modularization. https://github.com/jku-win-se/quantum_modularization (2021). Accessed June 2021
18. Gupta, S., Mittal, S., Gupta, T., Singhal, I., Khatri, B., Gupta, A.K., Kumar, N.: Parallel quantum-inspired evolutionary algorithms for community detection in social networks. Appl. Soft Comput. (2017). https://doi.org/10.1016/j.asoc.2017.07.035
19. Lancichinetti, A., Fortunato, S.: Community detection algorithms: a comparative analysis. Phys. Rev. E. Stat. Nonlinear Soft Matter Phys. (2009). https://doi.org/10.1103/PhysRevE.80.056117
20. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. Phys. Rev. E. Stat. Nonlinear Soft Matter Phys. (2008). https://doi.org/10.1103/PhysRevE.78.046110
21. Leymann, F., Barzen, J.: The bitter truth about gate-based quantum algorithms in the NISQ era. Quantum Science and Technology. 5(4) (2020)
22. Lotshaw, P.C., Humble, T.S., Herrman, R., Ostrowski, J., Siopsis, G.: Empirical performance bounds for quantum approximate optimization. arXiv:2102.06813. (2021)
23. Mann, H.B., Whitney, D.R.: On a test of whether one of two random variables is stochastically larger than the other. Ann. Math. Stat. (1947). https://doi.org/10.1214/aoms/1177730491
24. McGeoch, C., Pau, F.: Advantage Datasheet. https://dwavesys.com/resources/publications (2020). Accessed June 2021
25. McGeoch, C., Pau, F.: D-Wave Hybrid Solver Service + Advantage: Technology Update. https://www.dwavesys.com/hybrid-solver-service-advantage-technology-update-0 (2020). Accessed June 2021
26. McGeoch, C., Pau, F.: Hybrid Solver for Discrete Quadratic Models. https://www.dwavesys.com/hybrid-solver-discrete-quadratic-models-0 (2020). Accessed June 2021
27. Mniszewski, S.M., Dub, P.A., Tretiak, S., Anisimov, P.M., Zhang, Y., Negre, C.F.A.: Downfolding the Molecular Hamiltonian Matrix using Quantum Community Detection. arXiv:2011.14268. (2020)
28. Moody, D.: The Physics of Notations: toward a scientific basis for constructing visual notations in software engineering. IEEE Trans. Softw. Eng. (2009). https://doi.org/10.1109/TSE.2009.67
29. Negre, C.F.A., Ushijima-Mwesigwa, H., Mniszewski, S.M.: Detecting multiple communities using quantum annealing on the D-Wave system. PLoS ONE (2020). https://doi.org/10.1371/journal.pone.0227538
30. Reittu, H., Kotovirta, V., Leskelä, L., Rummukainen, H., Räty, T.: Towards analyzing large graphs with quantum annealing and quantum gate computers. arXiv:2006.16702. (2020)
31. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. Empirical Softw. Eng. (2009). https://doi.org/10.1007/s10664-008-9102-8

32. Shang, J., Liu, L., Li, X., Xie, F., Wu, C.: Epidemic spreading on complex networks with overlapping and non-overlapping community structure. Phys. A Stat. Mech. Appl. (2015). https://doi.org/10.1016/j.physa.2014.10.023
33. Shaydulin, R., Ushijima-Mwesigwa, H., Negre, C.F.A., Safro, I., Mniszewski, S.M., Alexeev, Y.: A hybrid approach for solving optimization problems on small quantum computers. Computer. (2019). https://doi.org/10.1109/MC.2019.2908942
34. Streif, M., Leib, M.: Training the quantum approximate optimization algorithm without access to a quantum processing unit. Quantum Sci Technol. (2020). https://doi.org/10.1088/2058-9565/ab8c2b
35. Ushijima-Mwesigwa, H., Shaydulin, R., Negre, C.F.A., Mniszewski, S.M., Alexeev, Y., Safro, I.: Multilevel Combinatorial Optimization Across Quantum Architectures. arXiv:1910.09985. (2019)
36. Yuanyuan, M., Xiyu, L.: Quantum inspired evolutionary algorithm for community detection in complex networks. Phys. Lett. A. (2018). https://doi.org/10.1016/j.physleta.2018.05.044
37. Zachary, W.W.: An information flow model for conflict and fission in small groups. J. Anthropol. Res. (1977). https://doi.org/10.1086/jar.33.4.3629752
38. Zhou, L., Wang, S.T., Choi, S., Pichler, H., Lukin, M.D.: Quantum approximate optimization algorithm: performance, mechanism, and implementation on near-term devices. Phys. Rev. X. (2020). https://doi.org/10.1103/PhysRevX.10.021067
39. Zhu, L., Tang, H.L., Barron, G.S., Calderon-Vargas, F.A., Mayhall, N.J., Barnes, E., Economou, S.E.: An adaptive quantum approximate optimization algorithm for solving combinatorial problems on a quantum computer. arXiv:2005.10258. (2020)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.