A Hybrid Approach for Solving Optimization Problems on Small Quantum Computers

Journal:	*Computer*
Manuscript ID	COMSI-2019-01-0011.R1
Manuscript Type:	SI: June 2019: Quantum Realism
Date Submitted by the Author:	n/a
Complete List of Authors:	Shaydulin, Ruslan; Clemson University College of Engineering and Sciences, School of Computing Ushijima-Mwesigwa, Hayato; Clemson University College of Engineering and Sciences, School of Computing Negre, Christian Francisco Andres; Los Alamos National Laboratory Safro, Ilya; Clemson University College of Engineering and Sciences, School of Computing Mniszewski, Susan; Los Alamos National Laboratory Alexeev, Yuri; Argonne National Laboratory
Keywords:	C.1.3.f Heterogeneous (hybrid) systems < C.1.3 Other Architecture Styles < C.1 Processor Architectures < C Computer Systems Organization, C.1.m.b Hybrid systems < C.1.m Miscellaneous < C.1 Processor Architectures < C Computer Systems Organization, E.1.d Graphs and networks < E.1 Data Structures < E Data, G.1.6.c Global optimization < G.1.6 Optimization < G.1 Numerical Analysis < G Mathematics of Computing

A Hybrid Approach for Solving Optimization Problems on Small Quantum Computers

Ruslan Shaydulin, Hayato Ushijima-Mwesigwa, Christian F. A. Negre, Ilya Safro, Susan M. Mniszewski, Yuri Alexeev

***Abstract*— Quantum architectures today possess major limitations that restrict the size of problems that can be solved directly. The ability to solve practical problems of increased size is an important area of research in the field of quantum computing. Designing hybrid quantum-classical algorithms is one of the most promising avenues. In this paper, we discuss decomposition-based hybrid approaches for solving optimization problems on small quantum computers and demonstrate them for two applications related to community detection in networks. The implementation details are demonstrated for two classes of noisy intermediate-scale quantum devices: the IBM universal quantum computer and the D-Wave quantum annealer.**

***Index Terms*—Quantum computing, universal quantum computing, quantum annealing, quantum–classical algorithms, variational hybrid algorithms, VQE, QAOA.**

I. Introduction

In recent years, quantum devices with up to tens of qubits on universal quantum computers (UQC) and a few thousand qubits on quantum annealer (QA) devices have become available. It enabled researchers to use real quantum hardware to solve "toy problems" for the first time. Unfortunately, in the near term, the devices are expected to stay very limited both in the number and the quality of qubits, making it hard to use these quantum computers for practical applications, which often require hundreds or even thousands of qubits. Such challenges as the qubit connectivity limitations, high level of noise, overhead of full error-correction, and concerns about scalability raise questions about the ability of near-term quantum hardware to effectively incorporate a larger number of qubits and deliver the theoretical speedups promised by many algorithms developed since 1990s.

Hybridization of quantum and classical algorithms is one of the expedient answers that researchers suggest today to tackle real-life problems with existing quantum hardware. These hybrid algorithms combine both classical and quantum computers in an attempt to take the advantage of "the best of both worlds", leveraging the power of quantum computation while using a classical machine to address the limitations of Noisy Intermediate-Scale Quantum (NISQ) computers (see Fig. 1). This is true not only for optimization algorithms, as discussed in this paper, but also for other problems, including quantum simulation [5], quantum machine learning [2], [12], [16] and more [10]. For example, classical computers have large memory and are capable of storing the entire global problem which is a challenge for NISQ devices with a small number of qubits. At the same time, quantum algorithms have shown improved performance for certain problems. To unambiguously distinguish between the stages of computation performed on two principally different types of hardware, we will refer to the classical and quantum stages of hybrid algorithms as CPU (including such accelerators as GPUs and FPGAs) and QPU (Quantum Processing Unit, including quantum annealer and a universal quantum computer), respectively.

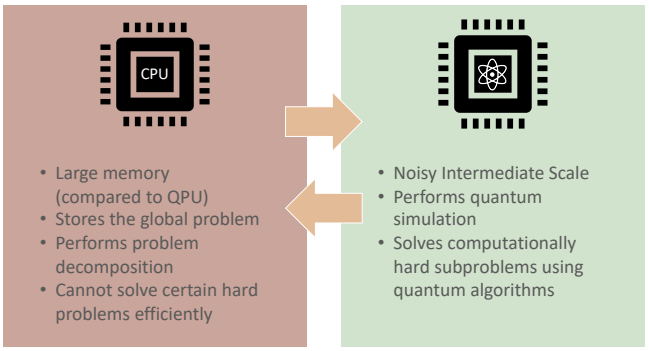

Fig. 1. Hybrid algorithms combine computations performed in both, a classical computer (CPU) and a NISQ quantum computer (QPU). Hybrid algorithms are designed with the goal of leveraging the strength of each mode of computation while dealing with its weaknesses. For example, CPUs cannot efficiently perform quantum simulation whereas modern small near-term QPUs cannot compute problems with many variables.

We primarily focus on two classes of NISQ-era devices, namely, UQCs and QAs, using IBM and D-Wave as exemplars. The IBM devices belong to the class of UQCs, which evolve the system by applying gates described using quantum assembly language (QASM). Other companies developing UQCs include Rigetti, Google, Microsoft, IonQ, and more. Alternatively, D-Wave devices are QAs designed to solve computational problems via quantum evolution towards the ground states of the cost Hamiltonians that encode optimization problems. While the two paradigms are very different, they share a lot of limitations and challenges.

Certain instances of graph partitioning [20] and community detection (also known as graph clustering) [11], [19] provide examples of practically important NP-hard optimization problems that can be solved by quantum computers today. A small

R. Shaydulin and I. Safro are with School of Computing, Clemson University, Clemson, SC 29634
H. Ushijima-Mwesigwa is with Fujitsu Laboratories of America, Sunnyvale, CA 94085 and previously affiliated with Clemson University.
C. F. A. Negre is with Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545.
S. M. Mniszewski is with Computer, Computational, & Statistical Sciences Division, Los Alamos National Laboratory, Los Alamos, NM 87545.
Y. Alexeev is with Computational Science Division, Argonne National Laboratory, Argonne, IL 60439
Manuscript received January 28, 2019; revised January 28, 2019.

graph problem can be solved directly on a QPU, while larger graph problems of practical interest require hybrid quantum-classical approaches.

As problems become too large to run directly on quantum computers, decomposition methods are required to split the problem into pieces of QPU-manageable size, an idea borrowed from high-performance computing and classical numerical methods. Static methods partition into subproblems that are run on the QPU and assemble all the pieces into a final solution on the CPU, while dynamic techniques use data-driven classical processing that produces subproblems that are run on a QPU as the solution evolves. Decomposition can be combined with compression or hierarchical approaches such as the multilevel method to better utilize a problem structure before decomposing it for a QPU. For examples of such schemes applied to hybrid quantum-classical algorithms the readers are referred to [3], [19].

The decomposition scheme we discuss is a natural evolution of variational quantum algorithms like Variational Quantum Eigensolver (VQE) and Quantum Approximate Optimization Algorithm (QAOA). The classical computer is not only used to find better parameters, but also finds an optimal subproblem size to solve on a quantum computer followed by an aggregation step. Given the size of current quantum computers this approach obviously does not allow for quantum speedup because it is limited to speedup achieved at the subproblem level. The fundamental difference with this classical scheme is that the subproblem size is chosen to maximize quantum speedup, which means running calculations on as many qubits as possible. Of course, two n-qubit quantum computers would be less powerful than one $2n$-qubit computer. When quantum hardware improves enough to be able to tackle a given problem *directly*, the decomposition methods might not be the best approach because of the overhead introduced. However, in the absence of $2n$-qubit computers decomposition-based methods provide a way to take advantage of the computational power of existing hardware.

In this overview, we do not focus on analyzing the performance and speedups afforded by quantum optimization algorithms like D-Wave quantum annealing and QAOA. For an overview of state-of-the-art on these methods we refer the reader to a recent report by the U. S. Department Of Energy's Office of Scientific and Technical Information [13], which shows that QAOA provably outperforms the best known classical approximation algorithm for special hard cases of the MAXCUT problem. Instead, we focus on a different question: if these methods *do* work, how can we use them to solve practical problems (i.e. find solutions of satisfactory quality) under the limitations of NISQ-era hardware? Assuming the methods can deliver speedups on problems that can fit on small near-term quantum computers, how can we leverage these speedups for solving problems of practical importance? All the decomposition-based methods described in this paper rely on the ability of quantum optimization methods to deliver quantum speedups on NISQ devices. Demonstration of quantum advantage on NISQ devices is currently an active area of research.

There are only a few frameworks today that provide an easy way to implement hybrid algorithms. Examples include the XACC framework [8] developed at ORNL for both UQCs and D-Wave QAs, Hybrid^{TM} workflow platform [1] by D-Wave, and Rigetti's Quantum Cloud Services (QCS) for UQCs. These frameworks follow the traditional co-processor model, i.e., treating QPUs as co-processors to execute specialized kernel code, while taking into account the complexity of interplay between classical and quantum hardware.

II. Overview of algorithmic approaches

In this section, we provide an overview of the QA and UQC paradigms and their specific hybrid methods for optimization. This is followed by a description of a general hybrid approach using the local search method, an optimization technique, solving subproblems on NISQ QPU devices (referring to both UQC and QA) with the main driving routine working on the CPU.

A. Quantum Annealing

QA devices are available today from D-Wave in the form of the 2X (up to 1152 qubits) and 2000Q (up to 2048 qubits). Using entanglement, they minimize the *Ising model* objective function composed of biases and strengths that encode the problem Hamiltonian. QAs are able to solve problems that arise in optimization, machine learning, sampling, and simulation. Problems formulated as a maximization are solved by the D-Wave using the negative of the objective function. In many occasions it is easier to use 0 and 1 as possible values for variables, leading to the problem formulated as a *quadratic unconstrained binary optimization* (QUBO).

Currently, D-Wave annealers have physical constraints such as limited precision, sparse connectivity, and number of available qubits. Problem variables do not map one-to-one with the available qubits. Each problem variable is represented by a chain of qubits obtained through embedding or mapping a problem onto the hardware *Chimera* graph prior to annealing. Quantum-only solutions are limited by the largest number of graph nodes/variables that can be represented on the D-Wave hardware. On the 2000Q, problems of up to size 64 can use the same embedding for 64 fully connected nodes/variables. Quantum-classical approaches are required for larger problems.

D-Wave's *qbsolv* tool is available for solving large QUBO problems using a hybrid quantum-classical algorithm [4]. In this case, dynamic decomposition is driven by an optimization solver. During each iteration of this algorithm a large neighborhood local search is performed, where subQUBOs are solved using the QPU, followed by tabu improvements on the CPU. The size of a subQUBO is limited by the number of variables that can be embedded in the hardware. The *qbsolv* loop converges to a low-energy solution.

B. Universal (gate-based) Quantum Computers

Near-term UQC devices are widely expected to have no more than a few hundred non-error-corrected qubits [14]. These NISQ computers cannot run many of the most famous

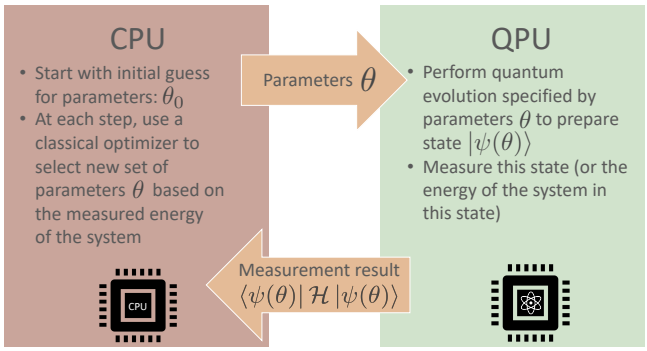

Fig. 2. General outline of variational hybrid algorithms. The variational cycle starts with some initial guess θ_0. Then at each step a trial state $|\psi(\theta)\rangle$ is prepared on the QPU (right), measured and the measurement result is read by the CPU (left). Then a classical optimization routine uses this measurement to select a next set of parameters θ and the cycle proceeds. This cycle continues until a solution of satisfactory quality is discovered or the classical optimizer converges.

QC algorithms with asymptotic speed-ups, like Shor's, for problems of practical size [14]. This is due to both the small number of qubits, as well as the limited number of gates that can be executed before the errors accumulate and the output becomes no longer useful due to decoherence. To address this challenge, a number of quantum-classical algorithms were introduced, most famous of them VQE and QAOA. QAOA can use either a variational or analytical strategy for finding parameters.

Variational algorithms combine a small QPU and CPU with the goal of finding the ground state of a problem Hamiltonian, which can be purely classical (QAOA) or quantum (VQE). An ansatz or trial state is prepared on the QPU and its energy is measured. The ansatz is prepared by applying a series of parameterized gates. This process is outlined in Fig. 2.

The advantage of variational algorithms is that the ansatz can be chosen such that the number of gates required is small enough to feasibly run on NISQ devices. This can introduce various trade-offs in terms of the quality of the trial state, complexity of the classical optimization and the amount of accumulated error. For example, in previous work [19], a recently introduced hardware-efficient ansatz [6] (see right part of Fig. 3) was used. This ansatz uses natural entangling interactions available on the device and therefore introduces less error; at the same time, it is much harder for a classical optimizer to find optimal parameters [9].

C. A General Hybrid Approach

As emerging NISQ devices are very limited in quality and number of qubits, they cannot tackle many problems of practical size directly. Variational algorithms address only the first part of the problem, namely the quality of the qubits. By using shallow-depth ansatzes, we can reduce the number of gates and bypass the issue of quickly accumulating errors. However, that still leaves the issue of problem size.

A natural way to address this challenge is to decompose the problem (statically or dynamically), solve the computationally hard subproblems on the QPU (either UQC or QA) and combine them on the CPU to obtain a global solution. Recently introduced Quantum Local Search (QLS) [18], [19] is one of the methods that utilize this approach. The QLS is inspired by the success of numerous local-search heuristics. The local-search methods have been applied to a variety of computationally hard problems (such as the satisfiability and traveling salesman problems) that otherwise are difficult for global solvers even in a reasonable amount of time. In QLS, the entire (global) problem is stored on the CPU. QLS starts from a random initial solution and at each step a neighborhood of the current solution is searched with the goal of improving the objective function. Since the neighborhood can be restricted to be small, this search can be performed on the QPU. If a better solution is found, the current solution is updated and QLS proceeds. This step is repeated until the objective function can no longer be improved. Similar approaches have been applied to mapping optimization problems to quantum annealers [3], [7], [17].

The advantage of QLS is that by storing the entire (potentially, very large) problem on CPU and only solving the computationally intensive *subproblems* on the QPU, we are able to tackle large problems using a limited hardware. Note that subproblems being offloaded to the QPU are not trivial; as larger quantum hardware becomes available, algorithms like QA and QAOA have potential to demonstrate quantum advantage on these subproblems. In [19], we were able to cluster graphs of up to 400 nodes using only a small 16 qubit QPU. However, this method has an obvious downside: it is prone to getting stuck in a local optima. If at a given iteration all neighborhoods we consider are too small to find an improvement, the algorithm would not be able to climb out of a local optima.

This issue is not unique to QLS. Generally, iterative local improvement methods are not sufficient to obtain high quality solutions for large problems unless combined with other global search algorithms. This limitation has led to the creation of multilevel (or multiscale) optimization where scale interactions of the given problem play an important role. The idea behind the multilevel method for solving large-scale problems is to create a hierarchy of smaller problems, which are easier to solve, and then work backwards towards the solution of the original problem, using a solution inherited at the coarser level of the hierarchy as an initializer of the next-finer level. The formation of the hierarchy therefore forms a basis to make global decisions for a given problem. More precisely, with respect to the graph partitioning problem, a graph is gradually coarsened to one where a partition can be computed efficiently and then this partition is projected back onto the original graph while being refined at all levels of the hierarchy. In the context of hybrid quantum-classical algorithms, the main driving routine that creates a hierarchy and assembles the final solution at all levels is performed on the CPU, and a costly refinement that solves subproblems is performed on the QPU.

III. Applications

After introducing the concepts and ideas related to the application of NISQ devices on large-scale problems, we focus

on two applications, namely, the network community detection for which we apply the quantum local search [18], [19] with both IBM Q and D-Wave architectures, and imidazole glycerol phosphate synthase (IGPS) protein sub-structure discovery using D-Wave and *qbsolv*.

A. Network community detection on IBM Q and D-Wave

The network community detection aims to group vertices based on their similarity which is often expressed in the number of shared immediate and distant neighbors between vertices. Such groupings frequently lead to dense and sparse link connectivity within and between the groups, respectively. Both densification and sparsification of inner and outer connections serve as objectives for a big class of community detection approaches. One of the most commonly used methods is modularity maximization.

In modularity maximization, the goal is to split the set of vertices of an underlying graph into two subsets (communities) such that the difference (called modularity) between the actual number of edges falling within a community and the expected number of edges if the edges were distributed uniformly at random over the same number of vertices is maximized. In other words, what modularity maximization looks for is a "statistically surprising" distribution of edges. In [19], we focus on finding an optimized assignment for two communities. However, there are several approaches to extend the problem to cases with more than 2 communities [11].

Solving community detection on NISQ computers is challenging because networks from practical applications are too large to be fully mapped onto the near-term hardware which justifies an application of QLS to address this issue. An outline is presented in Fig. 3.

QLS starts with some initial solution, which in the context of community detection is a community assignment. In [19], we start with a random community assignment, but in general it is possible to start with some other initial guess that is a result of a heuristic with acceptable time/quality trade-off.

At each step of QLS, a neighborhood of the current solution is explored. In two-way community detection, the solution space is all possible community assignments. Therefore a current solution's neighborhood is community assignments similar to it, i.e., those that only differ in a small number of vertices. Then the neighborhood search is performed at each step as follows. A subset of vertices to be moved between communities is selected based on some optimization criteria (in [19], we choose the highest gain change in the objective). The assignments of all other vertices remains invariant, and a new optimization subproblem is formulated by encoding these invariant vertices as boundary condition.

Note that by restricting the neighborhood size (i.e. number of vertices in the subset) we can reduce the number of variables in an optimization subproblem until it is small enough to fit on a NISQ device. This entails a tradeoff, since [19] shows that increasing the size of the neighborhood improves the convergence of the algorithm. But as new more capable quantum hardware becomes available, QLS has the potential to outperform classical state-of-the-art.

An additional benefit of QLS is that it is fundamentally hardware agnostic. As long as the subproblem can be mapped to a quantum algorithm executable on a given hardware, QLS can use it. In case of network community detection, the subproblem is in Ising form, which maps directly to both QA and QAOA. In [19], we demonstrate that QLS is able to find optimal solutions using 16 variable subproblems with both the IBM 16 qubit UQC device and the D-Wave 2000Q QA device. Note that subproblems of up to 64 fully connected variables can be solved on the D-Wave 2000Q using the same embedding. And the IBM 20 Q Tokyo, the largest UQC device available today, allows for up to 20 variable subproblems.

B. Multiple community detection using D-Wave

In this section we show an example of the applicability of the "all at once" (k-concurrent) community detection method as described in [11] for the D-Wave QA. In previous work we demonstrated that multiple community detection using the D-Wave QA leads to highly accurate community splits, that in most cases match the best known optimal solutions, particularly when graphs are small.

To show how useful this technique is, we picked an example corresponding to the community detection on dynamically correlated amino acid residues of the IGPS protein. This is an archetypal system that has been extensively studied in the field of biophysics [15]. The modularity matrix is calculated from the α-carbons pair-based correlation matrix that was obtained in [15] from a molecular dynamics (MD) simulation. The 454 amino acid residues of the IGPS protein form a graph with edge weights described by the correlation matrix.

The QUBO formulation for the global problem of up to k communities was created with $k \times n$ variables (where k is the maximum number of communities, and n is the number of vertices) and served as input to *qbsolv*. A subproblem size of 64 variables was used when running *qbsolv*. The split into two communities resulted in naturally revealing the two molecules that compose IGPS (see Figure 4). Solving for $k = 4$ communities resulted in identifying four domains (two per molecule) that might well be related to the protein functionality as shown in [15] (see Figure 4).

Albeit the results obtained with this example are not surprising, they serve to draw some useful conclusions. On the one hand, this demonstrates that quantum-classical algorithms using QAs applied to graph problems are moving out of the "toy model" exploratory phase. On the other hand, allowing for larger numbers of communities in the split, automatically leads to the need for having many more qubits available, hence justifying the need for hybrid quantum-classical algorithms.

IV. Conclusion and Outlook

In this paper, we discussed the hybrid quantum-classical approaches to quantum computing. Due to the limitations of NISQ QPU devices, using them as accelerators or co-processors for solving domain-specific problems in combination with a CPU is one of the most promising directions. Classical computers are used to prepare and post-process data obtained from quantum computations, but also in critical

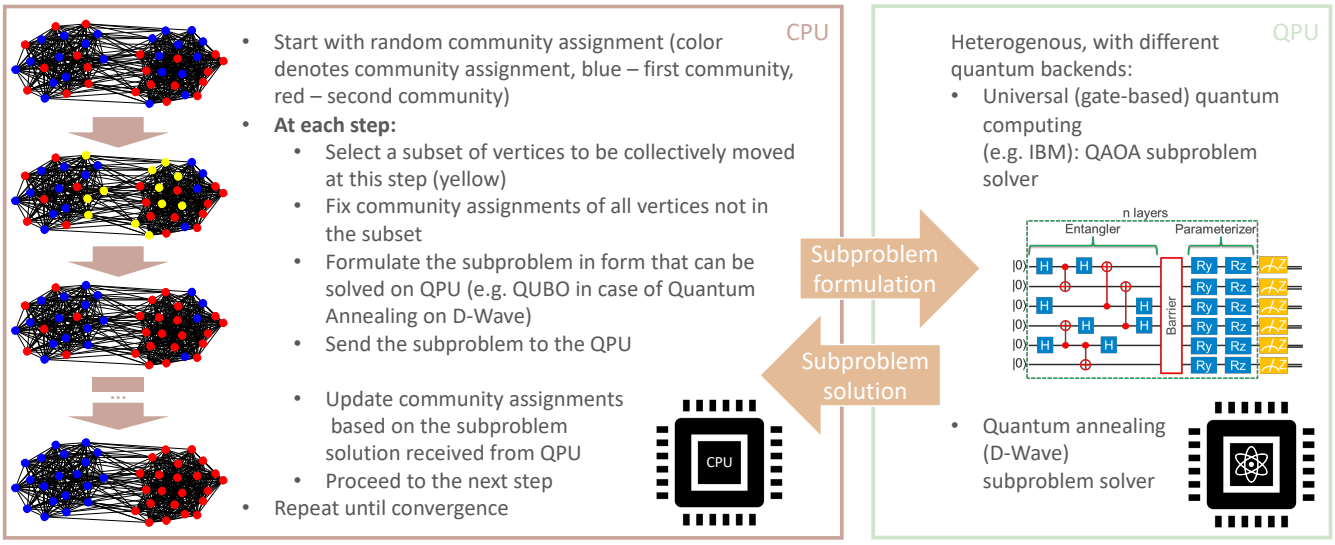

Fig. 3. An outline of Quantum Local Search applied to the problem of two-way network community detection (graph clustering with two clusters). Note that algorithms on the right (QPU box) can be hybrid in themselves, like QAOA described in Fig. 2. The set of gates presented on the right prepares the hardware-efficient ansatz [6] used in QAOA.

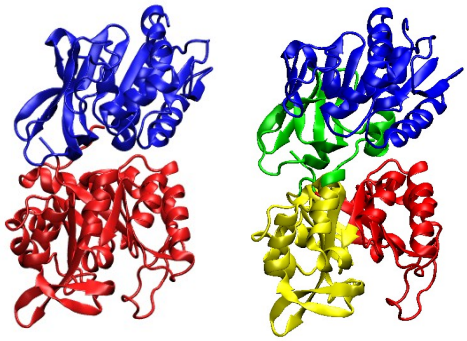

Fig. 4. The IGPS protein representation showing the split into two communities as its two different molecules (left) and four communities as two domains per molecule (right).

steps to find parameters for variational algorithms or produce subproblems using decomposition methods.

We demonstrated a few practical applications designed to work for such setups on IBM and D-Wave quantum computers. The discussed applications are all formulated as combinatorial optimization problems, but hybrid computing is not limited only to them. It is important to note that the hybrid quantum-classical approach will likely continue to be relevant even as quantum machines scale in the foreseeable future. First, there is no clear path to tens of thousands, not to mention millions of qubits. Second, it is clear that classical computers will remain better at certain things (for example, it is much more efficient to store large amounts of classical data on the classical computer), something that hybrid algorithms address by trying to leverage "the best of both worlds". Finally, there are always bigger and more complicated scientific problems to solve.

The hybrid decomposition-based approaches we advocate are not a silver bullet. They allow us to leverage small quantum computers to solve practical problems, but they are still limited by the capabilities of NISQ devices. If small quantum computers demonstrate quantum speedup, the decomposition-based methods will benefit from it and also show speedup for solving large optimization problems with modest overhead. If, on the other hand, there is no quantum speedup on NISQ devices, the decomposition methods will not be able to create it. The same considerations apply to other hybrid algorithms. If the "quantum" part of a quantum-classical algorithm cannot show any speedup on a NISQ device then hybridization by itself would not be able to provide it. There is no hybrid "free lunch".

We believe that the outlined major issues can be solved and that quantum computing has a promising future. We are currently seeing results that are comparable or better than *state of the art* classical methods in terms of quality of the solution, though runtime performance is lacking. Over the next 10 years, we expect improvements in qubits (quality, count and connectivity), error-correction, and quantum algorithms that will make improvements in runtime possible. Improvements in gate error rates will allow for more precise manipulation of qubits and enable more advanced computation. These will provide the missing ingredients for decomposition and hybrid quantum-classical methods, which rely on robust quantum computers capable of exhibiting quantum advantage. As the hardware evolves beyond NISQ era, we believe the decomposition methods will evolve with the capabilities of quantum devices.

V. Bios

Ruslan Shaydulin received his B.Sc. in Applied Mathematics and Physics from Moscow Institute of Physics and Technology and is currently pursuing Ph.D. in Computer Science at Clemson University. His work is focused on quantum and classical algorithms for combinatorial optimization and

machine learning. He has been affiliated and closely working with researchers from Argonne National Laboratory since 2018.

Hayato Ushijima-Mwesigwa received his Ph.D. in Computer Science from Clemson University. He joined Fujitsu Laboratories of America as a research scientist in February 2019 working on the design of optimization algorithms for quantum computers with applications to machine learning. He has also been affiliated and closely working with researchers from Los Alamos National Laboratory.

Christian Negre is a Staff Scientist at Los Alamos National Laboratory. He received his Ph.D. in Physical Chemistry from the National University of Córdoba, Argentina and later moved to the US for a postdoctoral position at Yale University, chemistry department. His main research area focuses on computational physical chemistry, including quantum dynamics, quantum molecular dynamics, and molecular electronics.

Ilya Safro is an Associate Professor of Computer Science and Faculty Scholar in School of Health Research at Clemson University. He received his Ph.D. in Applied Mathematics and Computer Science from Weizmann Institute of Science and was an Argonne Scholar at Argonne National Laboratory. His research interests include graph algorithms, machine learning, combinatorial scientific computing, and quantum-classical methods.

Susan Mniszewski is a Senior Scientist at Los Alamos National Laboratory. She received her B.Sc. in Computer Science from Illinois Institute of Technology. Her current research interests include quantum computing, neuromorphic computing, tensor networks, and quantum molecular dynamics on high performance computing architectures.

Yuri Alexeev is a Principal Project Specialist at the Argonne Argonne National Laboratory. He received his Ph.D. in Physical Chemistry from Iowa State University. His main research interests include computational quantum chemistry, computational biology, molecular dynamics, quantum computing, and high performance computing.

REFERENCES

[1] D-Wave releases hybrid workflow platform to build and run quantum hybrid applications in leap quantum application environment, December 2018. [Online; posted 10-December-2018].

[2] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195, 2017.

[3] Zhengbing Bian, Fabian Chudak, Robert Brian Israel, Brad Lackey, William G Macready, and Aidan Roy. Mapping constrained optimization problems to quantum annealing with application to fault diagnosis. *Frontiers in ICT*, 3:14, 2016.

[4] Michael Booth, Steven P Reinhardt, and Aidan Roy. Partitioning optimization problems for hybrid classical/quantum execution. Technical report, D-Wave systems Inc., 2017.

[5] Sergey Bravyi, Graeme Smith, and John A. Smolin. Trading classical and quantum computational resources. *Physical Review X*, 6(2):021043, 2016.

[6] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M Chow, and Jay M Gambetta. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature*, 549(7671):242, 2017.

[7] Hamed Karimi, Gili Rosenberg, and Helmut G Katzgraber. Effective optimization using sample persistence: A case study on quantum annealers and various monte carlo optimization methods. *Physical Review E*, 96(4):043312, 2017.

[8] Alexander J. McCaskey, Eugene F. Dumitrescu, Dmitry Liakh, Mengsu Chen, W. Feng, and Travis S. Humble. A language and hardware independent approach to quantum–classical computing. *SoftwareX*, 7:245 – 254, 2018.

[9] Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *arXiv preprint arXiv:1803.11173*, 2018.

[10] Jarrod R McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2):023023, 2016.

[11] Christian FA Negre, Hayato Ushijima-Mwesigwa, and Susan M Mniszewski. Detecting multiple communities using quantum annealing on the D-Wave system. 2018. (submitted).

[12] Daniel O'Malley, Velimir V. Vesselinov, Boian S. Alexandrov, and Ludmil B. Alexandrov. Nonnegative/binary matrix factorization with a D-Wave quantum annealer. *PLOS ONE*, 13(12):e0206653, 2018.

[13] Ojas D. Parekh, Ciaran Ryan-Anderson, and Sevag Gharibian. Quantum optimization and approximation algorithms. 1 2019.

[14] John Preskill. Quantum computing in the NISQ era and beyond. *Quantum*, 2(79), 2018.

[15] Ivan Rivalta, Mohammad M. Sultan, Ning-Shiuan Lee, Gregory A. Manley, J. Patrick Loria, and Victor S. Batista. Allosteric pathways in imidazole glycerol phosphate synthase. *Proceedings of the National Academy of Sciences*, 109(22):E1428–E1436, 2012.

[16] Jonathon Romero, Jonathon P. Olson, and Alan Aspuru-Guzik. Quantum autoencoders for efficient compression of quantum data. *Quantum Science and Technology*, 2(4):045001, 2017.

[17] Gili Rosenberg, Mohammad Vazifeh, Brad Woods, and Eldad Haber. Building an iterative heuristic solver for a quantum annealer. *Computational Optimization and Applications*, 65(3):845–869, 2016.

[18] Ruslan Shaydulin, Hayato Ushijima-Mwesigwa, Ilya Safro, Susan Mniszewski, and Yuri Alexeev. Community detection across emerging quantum architectures. *Proceedings of the 3rd International Workshop on Post Moore's Era Supercomputing*, 2018.

[19] Ruslan Shaydulin, Hayato Ushijima-Mwesigwa, Ilya Safro, Susan Mniszewski, and Yuri Alexeev. Network community detection on small quantum computers. *arXiv preprint arXiv:1810.12484*, 2018.

[20] Hayato Ushijima-Mwesigwa, Christian FA Negre, and Susan M Mniszewski. Graph partitioning using quantum annealing on the D-Wave system. In *Proceedings of the Second International Workshop on Post Moore's Era Supercomputing*, pages 22–29. ACM, 2017.