Hierarchical Improvement of Quantum Approximate Optimization Algorithm for Object Detection

(Invited Paper)

Junde Li, Mahabubul Alam, Abdullah Ash-Saki, and Swaroop Ghosh
School of Electrical Engineering and Computer Science, Pennsylvania State University, University Park, USA
Email: {*jul1512, mxa890, axs1251, szg212*}*@psu.edu*

***Abstract*—Quantum Approximate Optimization Algorithm (QAOA) provides approximate solution to combinatorial optimization problems. It encodes the cost function using a p-level quantum circuit where each level consists a problem Hamiltonian followed by a mixing Hamiltonian. Despite the promises, few real-world applications (besides the pedagogical MaxCut problem) have exploited QAOA. The success of QAOA relies on the classical optimizer, variational parameter setting, and quantum circuit design and compilation. In this study, we implement QAOA and analyze its performance for a broader Quadratic Unconstrained Binary Optimization (QUBO) formulation to solve real-word applications such as, partially occluded object detection problem. Furthermore, we analyze the effects of above influential factors on QAOA performance. We propose a 3-level improvement of hybrid quantum-classical optimization for object detection. We achieve more than 13X execution speedup by choosing L-BFGS-B as classical optimizer at the first level and 5.50X additional speedup by exploiting parameter symmetry and more than 1.23X acceleration using parameter regression at the second level. We empirically show that the circuit will achieve better fidelity by optimally rescheduling gate operations (especially for deeper circuits) at the third level.**

I. Introduction

Current Noisy Intermediate-Scale Quantum (NISQ) computers support dozens to hundreds of qubits. Quantum Approximate Optimization Algorithm (QAOA) [1] is a hybrid quantum-classical algorithm that exploits imperfect (and limited) number of qubits and integrates with classical routines for parameter optimization. For a p-level quantum circuit, the problem Hamiltonian H_C derived from classical cost function is applied p times, where each depth level i encompasses an additional variational parameter γ_i; while mixing Hamiltonian H_B is applied alternatively for p times as well. The mixing parameter is denoted by β_i. Variational parameters of QAOA $(\boldsymbol{\gamma}, \boldsymbol{\beta}) = (\gamma_1, \gamma_2, ..., \gamma_p, \beta_1, \beta_2, ..., \beta_p)$ are updated in each iteration of classical optimization and fed back into the p-level quantum circuit for generating quantum state $|\psi(\boldsymbol{\gamma}, \boldsymbol{\beta})\rangle$. The closed loop is repeated until the expectation value difference of H_C between two neighbouring iterations is within desired threshold.

The success of QAOA relies on classical optimizer, variational parameter setting strategy, quantum circuit design and compilation. However, the recent QAOA studies [2]–[6] are focused on MaxCut problem or its variants such as, *ring of disagrees*, which possibly fails to generalize its application space. In this work, we implement QAOA and analyze its performance on Quadratic Unconstrained Binary Optimization (QUBO) problem. Note that MaxCut is a special case of QUBO problem with only invariant coefficients and pair-wise binary terms. QUBO has wide applications in engineering and social sciences such as, vertex cover, clustering, and image thresholding, to name a few [7]–[9]. The QUBO model is defined with cost function: min/max $y = x^T Qx$, where x is a vector of n binary variables and Q is a nxn square matrix of variable coefficients in upper triangular form [8].

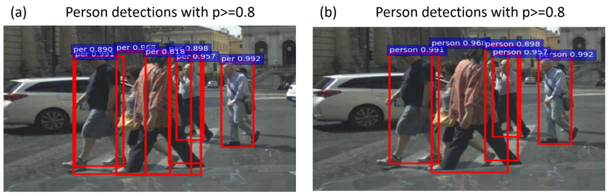


Fig. 1. Object detection using NMS and QUBO for removing false positive detection. Both methods are using confidence threshold of 0.8, (a) suppressing non-maximum detection locations from faster R-CNN with threshold of 0.5; (b) filtering out redundant locations from faster R-CNN using QUBO.

QAOA performance should be fairly evaluated and improved based on generic QUBO problems. To this end, we propose to improve the real-world application of object detection under partial occlusion by optimally filtering out redundant detection bounding boxes. Object detector generates overlapping bounding boxes, each of which has its own confidence value representing the likeliness of certain detected object. The number of raw detection boxes becomes the number of binary variables in QUBO model, and the detection probability formulates the diagonal elements in the Q matrix. The overlap ratio between two boxes helps to eliminate redundant locations. The boxes with higher overlap ratio are penalized in cost function. Thus a generic QUBO matrix Q is finally formulated with negative overlap ratio being the off-diagonal upper-triangular elements [10].

The existing object detection approaches almost exclusively use Non-Maximum Suppression (NMS) for removing redundant detection. Fig.1 shows detected objects using NMS and QUBO for filtering out redundant bounding boxes from Faster R-CNN detector [11]. It can be noted that QUBO suppresses redundancy further for this image instance though both are imperfect. This paper does not claim QUBO outperforms NMS from a general occluded object detection perspective. Rather, we use this real-world object detection problem to implement and analyze QAOA performance.

QAOA approximation ratio and execution time depend on classical optimization efficiency [12], variational parameter optimization [3], quantum circuit design and compilation [13]–[15]. Classical optimizers (both local and global) can be gradient-based, stochastic or heuristic. Their performances vary among optimization problems. QAOA p-level circuit with fixed parameters is regarded as sampling algorithm for certain number of quantum shots. However, finding optimal parameters $(\boldsymbol{\gamma}, \boldsymbol{\beta})$ is non-trivial, which requires either numerical analysis, parameter training or Hamiltonian symmetry exploitation. Optimal quantum circuit design reduces circuit delay by harnessing circuit parallelism, while efficient quantum algorithm compilation produces a sequence of analog control pulses with lower latency and gate error rates [15].

In this paper, we formulate a standard QUBO problem from real-world application of object detection under partial occlusion, based on which the performance of variational algorithm QAOA is analyzed from a generalized perspective. We further explore possible limiting factors on QAOA performance and propose hierarchical improvement by applying a suite of techniques. Our contributions are as follows:

- Evaluate QAOA performance on a standard QUBO model formulated by a real-world object detection problem instance;
- Compare the impact of various classical optimizers on QAOA performance, and identify the suitable technique for first-order improvement;
- Accelerate circuit execution by exploiting Hamiltonian symmetry to reduce parameter search space, and discuss other possible schemes for second-order improvement;
- Implement gate scheduling to leverage circuit parallelism for execution speedup for third-order improvement.

The rest of the paper is organized as follows: Section II provides background of QUBO model and QAOA algorithm. Section III describes QAOA implementation with various classical optimizers. Section IV explains variational parameter optimization, followed by quantum circuit optimization in Section V. Conclusion are drawn in Section VI.

II. Background

A. Quadratic Unconstrained Binary Optimization

Combinatorial optimization (CO) is used to solve many real-world applications [16], while QUBO serves as an unifying framework for solving CO problems [7]. Quadratic binary optimization is the problem of finding a binary string vector $\boldsymbol{x} = (x_1, x_2, ..., x_n)^T$ that maximizes the objective function $C(\boldsymbol{x})$. The standard objective function is typically formulated by linear and quadratic terms as below

$$C(\boldsymbol{x}) = \sum_{i=1}^{n} \sum_{j=i}^{n} c_{ij} x_i x_j + \sum_{j=1}^{n} c_i x_i = \boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} \qquad (1)$$

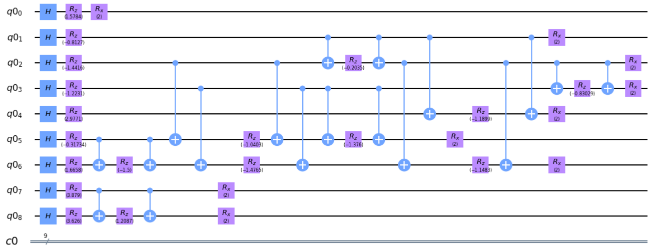
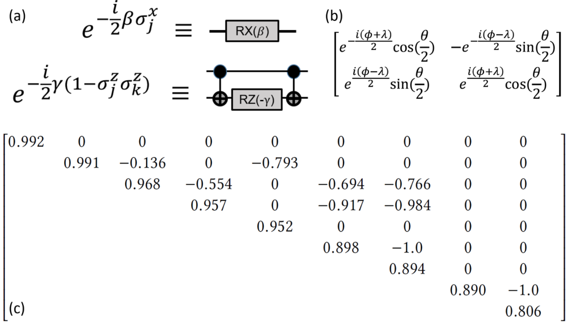


Fig. 2. Quantum circuit diagram of 1-level QAOA circuit for image instance in Fig. 1 with 9 qubits using Qiskit. H gate means qubit superposition initialization, one-qubit gate in H_C is denoted by $R_z(h_i\gamma_i)$ for qubit i, two-qubit ZZ interaction in H_C is denoted by two CNOT gates, plus a local one-qubit gate $R_z(J_{ij}\gamma_i)$ for qubits i and j, and each Pauli-X interaction in H_B is implemented with a single one-qubit $R_x(\beta_i)$ for qubit i.



$$e^{-\frac{i}{2}\beta\sigma_j^x} \equiv \qquad e^{-\frac{i}{2}\gamma(1-\sigma_j^z\sigma_k^z)} \equiv$$

$$\begin{bmatrix} e^{-\frac{i(\phi+\lambda)}{2}}\cos(\frac{\theta}{2}) & -e^{-\frac{i(\phi-\lambda)}{2}}\sin(\frac{\theta}{2}) \\ e^{\frac{i(\phi-\lambda)}{2}}\sin(\frac{\theta}{2}) & e^{\frac{i(\phi+\lambda)}{2}}\cos(\frac{\theta}{2}) \end{bmatrix}$$

$$\begin{bmatrix} 0.992 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & 0.991 & -0.136 & 0 & -0.793 & 0 & 0 & 0 & 0 \\ & & 0.968 & -0.554 & 0 & -0.694 & -0.766 & 0 & 0 \\ & & & 0.957 & 0 & -0.917 & -0.984 & 0 & 0 \\ & & & & 0.952 & 0 & 0 & 0 & 0 \\ & & & & & 0.898 & -1.0 & 0 & 0 \\ & & & & & & 0.894 & 0 & 0 \\ & & & & & & & 0.890 & -1.0 \\ & & & & & & & & 0.806 \end{bmatrix}$$

Fig. 3. (a) Unitary Pauli-X and Pauli-Z matrices in H_B and H_C respectively; (b) standard $U_3(\theta, \phi, \lambda)$ IBM logical basic gate for R_x operation; (c) upper triangular matrix Q representing target QUBO problem.

where $\forall i, x_i \in \{0, 1\}$ are binary variables, c_i are linear coefficients and c_{ij} are quadratic coefficients. Matrix Q represents the specific QUBO problem discussed in this study (Fig. 3(b)). It is a NP-hard problem to assign 0, 1 to each binary variable for maximizing the objective function. In the context of object detection problem, n denotes the number of detection bounding boxes, c_i denotes the detection probability, and c_{ij} denotes negative value of overlap ratio between two bounding boxes. Detection probability contributes to maximizing the objective function with positive c_i while overlap ratio is penalized by assigning its negative value to c_{ij}.

B. Ising Model

The classical optimization problem is mapped to quantum hardware by first encoding it into an Ising model. Then objective function is converted to $C(\boldsymbol{z}) = \boldsymbol{z}^T \boldsymbol{Q_{Ising}} \boldsymbol{z}$ for optimizing objective function $C(\boldsymbol{z}) : \{+1, -1\}^n \mapsto \mathbb{R}_{\geq 0}$ using Eq. 2 (refer to [17] for details).

$$C(\boldsymbol{z}) = \sum_{i=1}^{n} \sum_{j=i}^{n} J_{ij} z_i z_j + \sum_{j=1}^{n} h_i z_i = \boldsymbol{z}^T \boldsymbol{Q_{Ising}} \boldsymbol{z} \qquad (2)$$

The goal of quantum hardware is to optimize the string $\boldsymbol{z}^* = (z_1^*, z_2^*, ..., z_n^*)$ to achieve desired approximation ratio $\frac{C(\boldsymbol{z}^*)}{C_{max}} \geq r^*$, where $C_{max} = max_{\boldsymbol{z}} C(\boldsymbol{z})$ [2].

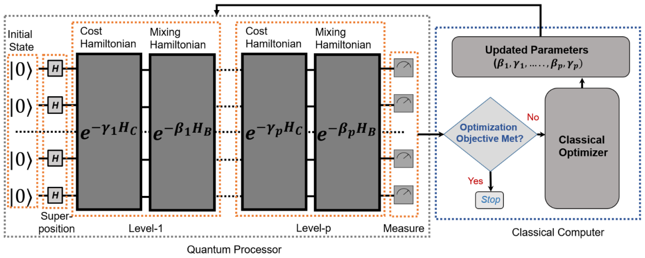


Fig. 4. Hybrid quantum-classical optimization algorithm for solving combinatorial problem.

C. QAOA Circuit

Fig. 2 shows the 1-level QAOA circuit of the image instance in Fig. 1. Nine qubits represent the number of bounding boxes (i.e., binary variables in cost function) before solved by QUBO. Values of each R_x and R_z gate reflect the product of coefficients in Q_{Ising} and circuit parameters. Note that h_i and J_{ij} are from coefficients of linear and quadratic terms of Q_{Ising}. Single qubit Pauli-X operation (R_x implemented by U_3 gate of Fig. 3(b)) and two-qubit ZZ interations(R_z) are shown in Fig.3(a). Qubit state measurement is not shown due to space constraint.

QAOA ansatz of level-p just iteratively alternates between applying each problem Hamiltonian H_C and mixing Hamiltonian H_B for p times. Quantum state is transferred after applying each unitary matrix operation. Problem Hamiltonian H_C is formed by replacing each binary variable $z_i \in \{+1, -1\}$ with a quantum spin of spin-1/2 σ_i^z: $H_C = C(\sigma_1^z, \sigma_2^z, ..., \sigma_n^z)$, and mixing Hamiltonian H_B is the transverse field $H_B = \sum_i \sigma_i^x$. Unitary matrix operators with gate parameters are represented below for H_C and H_B:

$$U_C(\gamma_i) = exp[-i\gamma_i H_C] \tag{3}$$

$$U_B(\beta_i) = exp[-i\beta_i H_B] \tag{4}$$

The standard initial quantum state $|\psi_0\rangle$ is in superposition state for all qubits by applying Hadamard gate. The corresponding density matrix for superposition state is as follows:

$$\rho_0 = |\psi_0\rangle \langle\psi_0| = \bigotimes_i \frac{1}{2}(\mathbb{1} + \sigma_i^x) \tag{5}$$

After alternatively applying unitary matrices, a p-level QAOA circuit is thus denoted by Eq. 6, and quantum state is converted from initial superposition state to final state as shown in Eq. 7.

$$U = U_B(\beta_p)U_C(\gamma_p)...U_B(\beta_2)U_C(\gamma_2)U_B(\beta_1)U_C(\gamma_1) \tag{6}$$

$$|\psi(\boldsymbol{\gamma}, \boldsymbol{\beta})\rangle = U\,|\psi_0\rangle \tag{7}$$

Further, the expectation value of H_C is determined in the variational state in Eq. 8. The performance of QAOA is measured by benchmarking merit of approximation ratio (AR): $r = F/C_{max}$. The role of classical computer in the hybrid quantum-classical method is to output optimal parameters $(\boldsymbol{\gamma}^*, \boldsymbol{\beta}^*)$ that maximizes $F(\boldsymbol{\gamma}^*, \boldsymbol{\beta}^*)$.

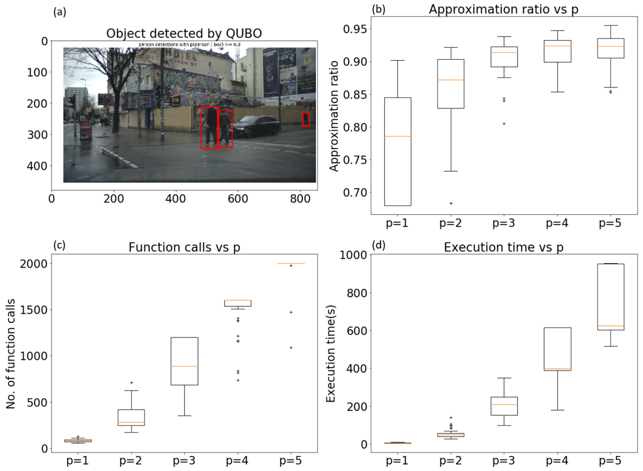


Fig. 5. QAOA performance metrics with increasing circuit depth p, (a) another image instance with lower number of bounding boxes for easier convergence; (b) approximation ratio of expectation value to C_{max}; (c) quantum function calls with Nelder-Mead; (d) hybrid quantum-classical algorithm execution time (time elapsed on Intel i7-level CPU).

$$F(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \langle\psi(\boldsymbol{\gamma}, \boldsymbol{\beta})|\, H_C \,|\psi(\boldsymbol{\gamma}, \boldsymbol{\beta})\rangle = \mathrm{Tr}\big[H_C U \rho_0 U^\dagger\big] \tag{8}$$

The iteration of hybrid quantum-classical circuit terminates until expectation value difference $\Delta F(\boldsymbol{\gamma}, \boldsymbol{\beta})$ between two neighboring iterations is within terminating threshold $\epsilon = 1e^{-4}$. At this moment, the best approximation ratio achieved is $r^* = F(\boldsymbol{\gamma}^*, \boldsymbol{\beta}^*)/C_{max}$. Besides, the number of iterations (a.k.a. quantum function calls) is another criteria for judging quantum approximation algorithm performance as it indirectly measures the circuit physical runtime.

III. Effects of Classical Optimizer

Based on basis state probability distribution, quantum circuit parameters are updated in each iteration of hybrid quantum-classical optimization. The update is classical optimizer-dependent, i.e. through local search or global search, gradient-based or gradient-free. We first evaluate the QAOA performance for another image with lower number of bounding boxes using a widely used classical optimizer Nelder-Mead (gradient-free local optimizer) [18]. The approximation ratio, quantum system function calls, and algorithm runtime with increasing depth from $p = 1$ to 5 are plotted in Fig. 5.

Without considering quantum gate errors and decoherence effects (ideal simulation), approximation ratio monotonically increases with increasing p, theoretically achieving $r \to 1$ when $p \to \infty$ because of higher number of dimensions of quantum Hilbert space for a higher level quantum circuit [19]. We run 20 instances with different seeds for the single image in Fig. 1, median AR achieved for $p = 5$ is 92.33%, lower than

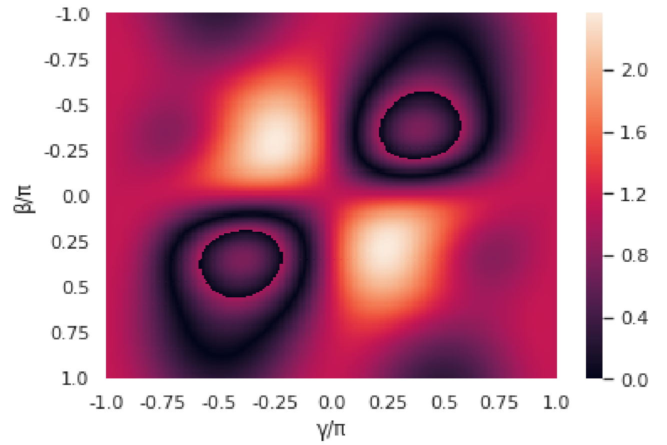


Fig. 6. Landscape of expectation value for quantum control parameters over entire solution space. It is obvious that $F(\boldsymbol{\gamma}, \boldsymbol{\beta}) = F(-\boldsymbol{\gamma}, -\boldsymbol{\beta})$.

the median 92.41% for $p = 4$. Since QAOA is simulated ideally, the only reason for such phenomenon is due to increased number of local optima suffered by Nelder-Mead for solution space with larger dimensions. Considering the descent AR of 92.41%, it's not worth running sufficiently many instances to guarantee monotonic AR increase for quantum circuit with higher depth. Landscape of expectation value as a function of (γ_1, β_1) within entire solution space $(-\pi, \pi)$ is visualized in Fig. 6 for 1-level QAOA ansatz.

Since AR may not increase for 5-level circuit with local optimizer, we examine the effects of classical optimizers on QAOA performance for circuit depth from $p = 1$ to 4. Aside from Nelder-Mead, we pick another two common optimizers from Scipy package [20], namely L-BFGS-B (gradient-based local optimizer) and Differential Evolution (stochastic global optimizer). The effects of classical optimizers are summarized in Table I.

TABLE I
QAOA PERFORMANCE COMPARISON WITH DIFFERENT CLASSICAL OPTIMIZERS (NM: NELDER-MEAD, LBB:L-BFGS-B, DE: DIFFERENTIAL EVOLUTION, AR: APPROXIMATION RATIO, FC:FUNCTION CALLS, TIME: ALGORITHM RUNTIME).

Opt.	p	AR	AR	FC	FC	Time	Time
		Mean	SD	Mean	SD	Mean	SD
NM	1	0.835	4.0E-11	64.5	4.73	78.62	6.18
	2	0.838	1.8E-11	201.25	14.03	486.34	38.47
	3	0.838	1.9E-11	717	75.40	2556.4	271.1
	4	0.849	0.009	1511.7	153	7171.8	757.6
LBB	1	0.835	2.0E-09	18	0	22.87	1.21
	2	0.838	1.1E-09	75	4.08	180.66	10.77
	3	0.838	2.5E-04	73.75	23.22	263.18	84.51
	4	0.838	2.0E-04	112.5	11.62	535.52	55.80
DE	1	0.835	1.4E-13	262.5	28.72	321.83	39.55
	2	0.838	5.0E-11	961.25	176.09	2333.1	463.9
	3	0.860	0.004	4440.5	654.16	15580	2170
	4	0.870	0.001	5333.3	973.28	25380	4727

Every classical optimizer runs the same target image (in Fig. 1) for 5 times, and the mean and standard deviation for AR, function calls, and physical runtime are calculated.

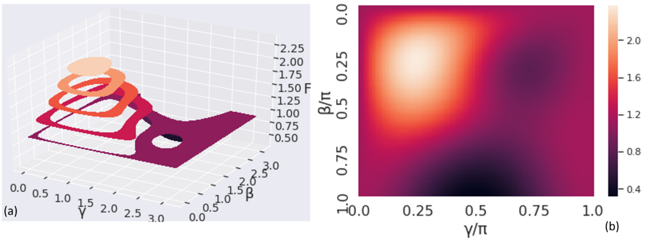


Fig. 7. (a) Contour plot of expectation value as function of (γ_1, β_1); (b) global optima is found at around $(\pi/4, \pi/4)$ shown in heatmap. No local maxima is observed in constrained space.

Approximation ratio is within range $(0.835, 0.870)$ for all classical optimizers with all 4-level circuit depth. Differential evolution achieves highest approximation ratio (increasing monotonically with circuit depth). However it takes around 35X more function calls (FC) than Nelder-Mead, and 47X more FCs than L-BFGS-B. In the following study, all experiments are performed with L-BFGS-B as it runs much faster than the other two and achieves comparable accuracy.

IV. Optimizing Variational Parameters

QAOA p-level circuit with well-constrained parameters accelerates hybrid optimization process by exploring fewer Hilbert solution space. When all gate parameters are fixed, QAOA works only as a sampler for measuring quantum basis state distribution. Therefore, this section exploits parameter patterns for constraining parameter exploration space.

A. Numerical Experiment

QAOA has a time-reversal symmetry [2]. Therefore, generically solution space is simplified by $F(\boldsymbol{\gamma}, \boldsymbol{\beta}) = F(-\boldsymbol{\gamma}, -\boldsymbol{\beta})$ [3]. Such numerical symmetry exploitation is supported by expectation value landscape (Fig. 6). Fig. 7 shows expectation value landscape using heatmap and contour map after constraining parameters from $(-\pi, \pi)$ to $(0, \pi)$ by eliminating degeneracies in Hilbert space. In order to find further parameter patterns, we repeat the numerical implementation for 20 other images (all of them have 6 bounding boxes) with constrained parameter range $(0, \pi)$ using only L-BFGS-B, the suitable one found in Section III. Each image is implemented with circuit depth from $p = 1$ to 3, as 3-level QAOA circuit achieves satisfying approximation ratio. The effects of constraining circuit parameter to $(0, \pi)$ is shown in Fig. 8 for visualizing the second-order improvement. Algorithm runtime is accelerated for 3.97X, 2.88X, and 5.50X for $p = 1$ to 3, respectively after exploiting parameter symmetry.

Parameter pattern is illustrated by simultaneously plotting optimal or near-optimal (due to local optima) gate parameters for all 20 images with 3 circuit depths, as shown in Fig. 9. Optimal H_C parameter γ^* tends to increase with larger circuit depth, while optimal H_B parameter β^* tends to decrease, which is consistent with the findings from [2]. We perform numerical implementation for each image for only one instance to visualize effects of local maxima in the entire

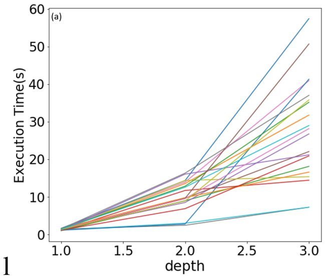


Fig. 8. (a) Algorithm runtime for 20 different images with solution search in entire Hilbert space $(-\pi, \pi)$; (b) circuit execution acceleration with constrained parameters in half space $(0, \pi)$.

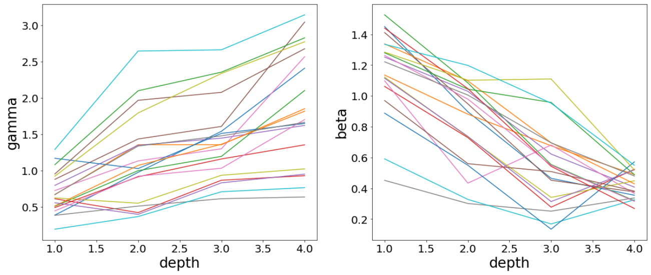


Fig. 9. Optimal or quasi-optimal QAOA parameters (γ^*, β^*) for 20 other images. Each image is repeated once for visualizing effects of local optima. Image instances with turning point of parameter curve are likely to suffer from local optima.

solution space. The likelihood of local optima is around 10% at 3-level circuit depth for generic QUBO problem.

B. Parameter Regression

Based on parameter patterns shown in Fig. 9, regression model is naturally exploited for predicting gate parameters of the next level depth. Since training data (parameters from 20 images) is generated from images with 6 binary variables, we examine possible performance acceleration for another image with 6 binary variables and three other images with 7-9 binary variables using multivariate linear regression. Note that, we only demonstrate that regression helps QAOA parameter optimization. More powerful regressors such as, Neural Network can be further explored with amounts of training data for handling gate parameters with large depth. Level-4 gate control parameters are predicted based on previous level optimized parameters. Trained parameters are fed into QAOA circuit for execution, and QAOA circuit with randomly initialized parameters are also implemented for 5 times for comparison. The acceleration results using parameter regression are shown in Table II. Average number of function calls over experiments with random parameters for these four type of image instances are 537, 230, 462, and 389, respectively. A 4.88X speedup is achieved for 6-qubit image compared to trained parameters-based approach.

TABLE II
AVERAGE ACCELERATION OF QUANTUM CIRCUIT IN TERMS OF FUNCTION CALLS.

Image	rand_1	rand_2	rand_3	rand_4	rand_5	trained	Acc.
6-qubit	649	517	836	506	176	110	4.88
7-qubit	213	198	143	319	275	187	1.23
8-qubit	605	165	748	483	308	275	1.68
9-qubit	406	220	259	550	506	209	1.86

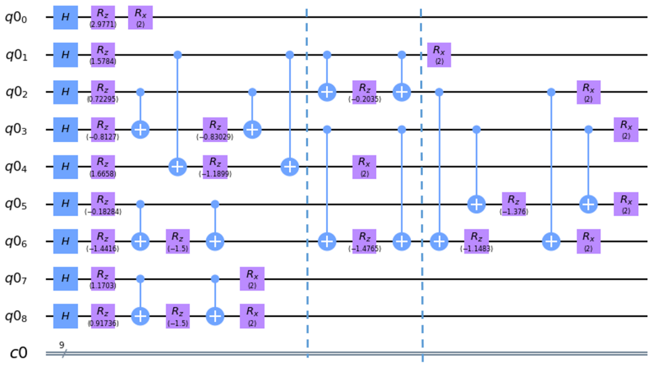


Fig. 10. Two-qubit ZZ interactions are classified into 4 layers after rescheduling, where 4 two-qubit interactions run in parallel in first layer compared to 5 layers shown in Fig. 2. $Qubit_{(2,3)}$, and $Qubit_{(1,4)}$ are non-conflicting though they are not shown in parallel in this diagram (pair $(2, 5)$ in the last layer is not shown).

V. OPTIMIZING CIRCUIT DESIGN AND COMPILATION

A. Gate Scheduling

Gate operation scheduling optimization harnesses more parallelism on the execution of the logical quantum gates. Given the same number of gate operations, QAOA circuit is executed faster due to a shorter critical path (the qubit with most unitary matrix transformations) with optimal rescheduling. Less physical runtime also mitigates the impacts of decoherence induced errors on the fidelity of cost Hamiltonian. Based on Eq. 6, the gate sequence can only be reordered among gates that are commutative. The non-commuting unitary matrix transformations, if computed with rescheduling, will generate an incorrect functionality. The constraint that any qubit is involved in at most once at a time has to be complied with during rescheduling [21]. QAOA performance improvement of this level is also demonstrated with target image in Fig. 1 by optimal gate scheduling. We manually reschedule the quantum circuit in Fig. 2 to generate the optimized one shown in Fig. 10. Rescheduling is only performed within cost Hamiltonian as gates between H_C and H_B are not commutative.

Gate rescheduling is performed using quantum computation simulator for better controlling gate and decoherence errors. We use the simulator developed by [22] with standard gate noise data and relaxation (T1 decoherence) and dephasing (T2 decoherence) times reported from IBMQX4 [23]. QAOA for target image is first performed without gate error and decoherence errors to obtain density matrix ρ_{ideal} for calculating fidelity. Then QAOA with decoherence errors are performed for circuits without and with gate rescheduling (ρ_{nonsch} and

ρ_{resch}), respectively. No gate error is included for these three experiments as gate scheduling may only relate to circuit runtime and decoherence errors. Fidelity between two density matrices is calculated using Eq. 9 [24].

$$Fid(\rho, \gamma) = \mathrm{Tr}\left[\sqrt{(\sqrt{\rho}\sqrt{\gamma})^{\dagger}(\sqrt{\rho}\sqrt{\gamma})}\right]^2 \quad (9)$$

Qubit relaxation time of 45us and dephasing time of 20us reported by [23] are modeled in [22]. The number of gate operations in critical path is 15, including pair-term (2, 5), as plotted in Fig. 10. QAOA circuit with depth $p = 1$ and 2 are implemented for testing fidelity improvement from gate rescheduling. However, the total gate operation time based on our simulator running on classical computer is higher by several orders of magnitude than that on quantum computer [23], making it infeasible to compare fidelity with ideal environment. Thus fidelity is only calculated between ρ_{nonsch} and ρ_{resch}. The results show that $Fid(\rho_{nonsch}, \rho_{resch})$ for level-1 circuit is 99.96% and 99.94% for level-2 circuit. Gate rescheduling improvement on circuit delay becomes more pronounced with increasing circuit depth.

B. Pulse Engineering

Another technique related to reducing program execution or compilation time is pulse engineering that produces the optimal possible sequence of analog control pulses for realizing unitary matrix transformations. Decreasing the quantum algorithm compilation time is critical to qubit infidelity due to decoherence effects. Partial compilation of variational algorithms that pre-computes optimal control pulses for parameter-independent gate operations has been proposed [15]. This strategy incurs no overhead for a dramatic compilation speedup in quantum algorithm.

VI. Conclusion

The performance of QAOA is empirically evaluated using QUBO framework in a generic scenario. Matrices such as, approximation ratio and function calls (together with execution time) obtained in this study benchmark the QAOA performance of QUBO problems. Our study shows the promise of object detection by hybrid quantum-classical approximation algorithm. Classical optimizer L-BFGS-B performed 10X faster than Nelder-Mead, and 47X faster than Differential Evolution at iso-approximation ratios. We shows that gate control parameter symmetry in Hilbert space accelerates physical runtime by 5.5X for level-3 circuit and expects to achieve higher acceleration for deeper QAOA circuit. Parameter pattern can be exploited by machine learning for predicting gate parameters. Gate operation rescheduling can harness more quantum computation parallelism, thereby reducing gate operation critical path and decoherence time.

References

[1] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," *arXiv preprint arXiv:1411.4028*, 2014.

[2] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin, "Quantum approximate optimization algorithm: performance, mechanism, and implementation on near-term devices," *arXiv preprint arXiv:1812.01041*, 2018.

[3] Z. Wang, S. Hadfield, Z. Jiang, and E. G. Rieffel, "Quantum approximate optimization algorithm for maxcut: A fermionic view," *Physical Review A*, vol. 97, no. 2, p. 022304, 2018.

[4] M. Alam, A. Ash-Saki, and S. Ghosh, "Analysis of quantum approximate optimization algorithm under realistic noise in superconducting qubits," *arXiv preprint arXiv:1907.09631*, 2019.

[5] C. Xue, Z.-Y. Chen, Y.-C. Wu, and G.-P. Guo, "Effects of quantum noise on quantum approximate optimization algorithm," *arXiv preprint arXiv:1909.02196*, 2019.

[6] G. G. Guerreschi and A. Matsuura, "Qaoa for max-cut requires hundreds of qubits for quantum speed-up," *Scientific reports*, vol. 9, no. 1, p. 6903, 2019.

[7] G. Tavares, "New algorithms for quadratic unconstrained binary optimization (qubo) with applications in engineering and social sciences," Ph.D. dissertation, Rutgers University-Graduate School-New Brunswick, 2008.

[8] F. Glover and G. Kochenberger, "A tutorial on formulating qubo models," *arXiv preprint arXiv:1811.11538*, 2018.

[9] V. Kumar, G. Bass, C. Tomlin, and J. Dulny, "Quantum annealing for combinatorial clustering," *Quantum Information Processing*, vol. 17, no. 2, p. 39, 2018.

[10] S. Rujikietgumjorn and R. T. Collins, "Optimized pedestrian detection for multiple and occluded people," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3690–3697.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[12] P.-I. Schneider, X. G. Santiago, V. Soltwisch, M. Hammerschmidt, S. Burger, and C. Rockstuhl, "Benchmarking five global optimization approaches for nano-optical shape optimization and parameter reconstruction," *arXiv preprint arXiv:1809.06674*, 2018.

[13] S. Khatri, R. LaRose, A. Poremba, L. Cincio, A. T. Sornborger, and P. J. Coles, "Quantum-assisted quantum compiling," *arXiv e-prints*, p. arXiv:1807.00800, Jul 2018.

[14] N. Leung, M. Abdelhafez, J. Koch, and D. Schuster, "Speedup for quantum optimal control from automatic differentiation based on graphics processing units," *Physical Review A*, vol. 95, no. 4, p. 042318, 2017.

[15] P. Gokhale, Y. Ding, T. Propson, C. Winkler, N. Leung, Y. Shi, D. I. Schuster, H. Hoffmann, and F. T. Chong, "Partial compilation of variational algorithms for noisy intermediate-scale quantum machines," in *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*. ACM, 2019, pp. 266–278.

[16] B. Korte and J. Vygen, "Combinatorial Optimization," 2012.

[17] N. Chancellor, "Domain wall encoding of discrete variables for quantum annealing and QAOA," *Quantum Science and Technology*, vol. 4, no. 4, p. 045004, aug 2019. [Online]. Available: https://doi.org/10.1088%2F2058-9565%2Fab33c2

[18] E. Farhi and A. W. Harrow, "Quantum supremacy through the quantum approximate optimization algorithm," *arXiv preprint arXiv:1602.07674*, 2016.

[19] E. Crosson, E. Farhi, C. Yen-Yu Lin, H.-H. Lin, and P. Shor, "Different Strategies for Optimization Using the Quantum Adiabatic Algorithm," *arXiv e-prints*, p. arXiv:1401.7320, Jan 2014.

[20] E. Jones, T. Oliphant, P. Peterson *et al.*, "SciPy: Open source scientific tools for Python," 2001–. [Online]. Available: http://www.scipy.org/

[21] G. G. Guerreschi and J. Park, "Two-step approach to scheduling quantum circuits," *Quantum Science and Technology*, vol. 3, no. 4, p. 045003, 2018.

[22] A. Ash-Saki, M. Alam, and S. Ghosh, "Qure: Qubit re-allocation in noisy intermediate-scale quantum computers," in *Proceedings of the 56th Annual Design Automation Conference 2019*. ACM, 2019, p. 141.

[23] "Ibm q experience," *https://quantumexperience.ng.bluemix.net/qx/*, [online].

[24] A. Y. Kitaev, A. H. Shen, and M. N. Vyalyi, *Classical and Quantum Computation*. Boston, MA, USA: American Mathematical Society, 2002.