Quantum Parametric Circuit Optimization with Estimation of Distribution Algorithms

Vicente P. Soloviev
vicente.perez.soloviev@fi.upm.es
Universidad Politécnica de Madrid
Madrid, Spain

Pedro Larrañaga
pedro.larranaga@fi.upm.es
Universidad Politécnica de Madrid
Madrid, Spain

Concha Bielza
mcbielza@fi.upm.es
Universidad Politécnica de Madrid
Madrid, Spain

ABSTRACT

Variational quantum algorithms (VQAs) offer some promising characteristics for carrying out optimization tasks in noisy intermediate-scale quantum devices. These algorithms aim to minimize a cost function by optimizing the parameters of a quantum parametric circuit. Thus, the overall performance of these algorithms, heavily depends on the classical optimizer which sets the parameters. In the last years, some gradient-based and gradient-free approaches have been applied to optimize the parameters of the quantum circuit. In this work, we follow the second approach and propose the use of estimation of distribution algorithms for the parameter optimization in a specific case of VQAs, the quantum approximate optimization algorithm. Our results show an statistically significant improvement of the cost function minimization compared to traditional optimizers.

CCS CONCEPTS

• **Computing methodologies** → **Concurrent computing methodologies**; • **Theory of computation** → **Design and analysis of algorithms**; • **Hardware** → **Quantum computation**.

KEYWORDS

Quantum optimization, variational quantum algorithm, estimation of distribution algorithm, max cut

ACM Reference Format:
Vicente P. Soloviev, Pedro Larrañaga, and Concha Bielza. 2022. Quantum Parametric Circuit Optimization with Estimation of Distribution Algorithms. In *Proceedings of The Genetic and Evolutionary Computation Conference 2022 (GECCO '22).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3520304.3533963

1 INTRODUCTION

In the last decades an exponential use of quantum hardware for different optimization tasks in the academic and industrial areas has been witnessed. Although it has been shown that the use of quantum hardware can outperform classical computation in some mathematical operations or in the energy consumption for some processes, it is still unclear if the use of noisy intermediate-scale quantum (NISQ) hardware can outperform classical computation for real use cases such as optimization.

There exist several research topics regarding quantum optimization. Some of the most relevant are quantum adiabatic computation (QA) and variational quantum algorithms (VQA) [7]. The former consists of translating the optimization problem into a set of Hamiltonians which define an energy function that describes the dynamics of a quantum system. This problem encoding is a quadratic unconstrained binary optimization (QUBO) problem, and several real world applications have been encoded into QUBOs [14]. The latter consists of translating the cost function to be optimized into a quantum state which is represented as a quantum parametric circuit, which in this work will be henceforth named as an *ansatz*. These parameters are iteratively optimized to minimize a cost function. A wide range of problems have been faced using VQA approaches [19, 21], due to multiple advantages such as its quantitative resilience to the quantum noise in NISQ devices [18].

VQAs are hybrid approaches where a classical optimizer iteratively proposes new sets of parameters to fit the *ansatz* in order to reduce the energy of the system. Thus, the overall performance of VQAs heavily depends on the performance of the classical optimizers. Barren plateau problem [15] is a particular challenge of this optimization problem,a phenomenon where gradients of the VQA parameters vanish exponentially with the number of qubits. Finding limitations of these optimizers is crucial depending on the task or the VQA complexity that is being used. Several studies have been carried out comparing traditional optimizers in this problem [5]. VQAs are commonly known as quantum heuristics for sharing characteristics with classical heuristics such as population-based algorithms [2]. A well-known VQA is the quantum approximate optimization algorithm (QAOA) [10].

In this work, we propose to use estimation of distribution algorithms (EDAs) [13] to approach the *ansatz* parameter optimization. EDAs have demonstrated to achieve very good results in continuous space optimization for a wide range of problems [8], and thus, we believe that these algorithms can provide competitive results to approach this problem and overcome limitations such as the Barren plateau problem [15].

2 VARIATIONAL QUANTUM ALGORITHMS

The VQA objective is to find the approximate ground state of an n-qubit quantum system as an output to an *ansatz*. The parameters $\boldsymbol{\theta} \in [0, 2\pi]^d$ defined in the *ansatz* represents rotation of single or two-qubit gates. During VQA runtime, the classical optimizer proposes new sets of these vector $\boldsymbol{\theta}$ of parameters with d components. After setting the parameters, the quantum circuit is measured t times and the expectation value among solutions is computed. Since the aim of

GECCO '22, July 9–13, 2022, Boston, USA

ACM ISBN 978-1-4503-9268-6/22/07…$15.00
https://doi.org/10.1145/3520304.3533963

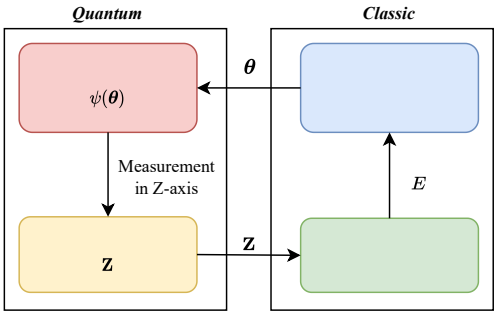

Figure 1: VQA flowchart for a hybrid approach between quantum and classical computing. In each iteration, a classical optimizer proposes a new set of parameters, and a measurement is performed in the *ansatz* ($\psi(\boldsymbol{\theta})$). Considering the computed expectation value, the classical optimizer proposes a new set of parameters. The algorithm iteratively runs until a convergence criterion is met.

VQA is to minimize the defined cost function, the aim of the classical optimizer is to minimize the expectation value computed in each iteration. Minimizing the expectation value will lead to cost function minimization. The general workflow is shown in Figure 1. Such an expectation can be obtained by performing measurements along the Z-axis of the quantum system and computing the following expression

$$E(Z_i) = \frac{1}{t} \sum_{z \in Z} C(z) N_{Z_i}, \tag{1}$$

where N_{Z_i} is the number of times solution z is measured by executing the circuit t times in iteration i, Z is the set of possible basis states obtained by the circuit, $C(z)$ is the evaluation of the solution z in the classical cost function to be optimized by the VQA, and Z_i the set of measurements obtained in iteration i.

2.1 Quantum Approximate Optimization Algorithm

QAOA was proposed by Farhi, Goldstone and Gutmann [10] for solving combinatorial optimization problems. The QAOA *ansatz* consists of $p \in \mathbb{N}$ layers, and each layer is formed by two different operators that encode the cost function to be optimized (Fig. 2) where the cost operator $U(H_C, \gamma)$ parameterized by $\gamma \in [0, 2\pi]$ has the following expression:

$$U(H_C, \gamma) = e^{-i\gamma H_C} = \prod_{\alpha=1}^{m} e^{-i\gamma C_\alpha} \tag{2}$$

where C_α is the cost function to be minimized, and the mixed operator $U(H_B, \beta)$ is parameterized by $\beta \in [0, 2\pi]$,

$$U(H_B, \beta) = e^{-i\beta H_B} = \prod_{j=1}^{n} e^{-i\beta \sigma_j^x} \tag{3}$$

where σ_j^x is the rotation in the X-axis of the qubits {+1, -1}.

The QAOA *ansatz* is represented as:

$$\psi(\boldsymbol{\gamma}, \boldsymbol{\beta}) = U(H_B, \beta_p) U(H_C, \gamma_p) \dots U(H_B, \beta_1) U(H_C, \gamma_1) \langle s \rangle$$

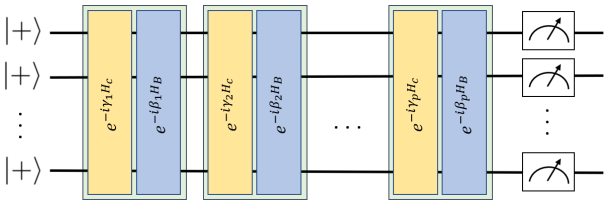

Figure 2: A quantum parametric circuit with p layers and $2p$ parameters. The initial state is a superposition of all the possible computational states, and after applying the p layers, a measurement along the Z axis of all the qubits is performed.

where $p \geq 1$, $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_p)$, $\boldsymbol{\beta} = (\beta_1, \dots, \beta_p)$, and $\langle s \rangle$ is the uniform superposition state over all possible computational states. A quantum circuit is composed by p layers and a total of $2p$ parameters ($\boldsymbol{\theta} = [\gamma_1, \beta_1, \dots, \gamma_p, \beta_p]$) to be optimized.

2.2 Parameter Optimization

From an optimization perspective, the *ansatz* parameter optimization task does not much differ from any other continuous optimization problem. However, in the NISQ era, several facts should be considered such as the number of layers to be optimized, as it defines the number of parameters to be optimized, or whether it is more important to achieve a better expectation value or to minimize the computation time.

Depending on whether the optimizers are gradient-based or gradient-free [3] we list the following algorithms commonly used for *ansatz* parameter optimization, which will be compared to the EDA approach in Section 4. Gradient-based optimizers include ADAM [12], conjugate gradient method (CG) [11], gradient descent [17], limited-memory Broyden–Fletcher–Goldfarb-Shanno algorithm (L-BFGS) [6] and sequential least squares programming (SLSQP) [4]. Gradient-free optimizers include constrained optimization by linear approximation (COBYLA) [16] and simultaneous perturbation stochastic approximation (SPSA) [20].

3 ESTIMATION OF DISTRIBUTION ALGORITHM

Evolutionary computation is one of the main branches in population-based algorithms [2]. Evolutionary algorithms are increasingly demanded for solving optimization problems as they can achieve very good results across a wide range of problem domains [8]. EDAs are one of the best known EAs. The main characteristic of EDAs is the generation of new solutions from a probabilistic model built from the best solutions of previous iterations.

Each solution considered during runtime is a set of parameters to be set into the *ansatz*. Thus, a solution is a vector defined as $\boldsymbol{\theta}^* = [\theta_1, \theta_2, \dots, \theta_d]$ where d is the number of parameters to be optimized in the *ansatz*.

The pseudocode of the EDA approach is presented in Algorithm 1 where E is the expectation value to be minimized (Eq. 1).

Depending on the problem characteristics, different probabilistic models can be considered in the EDA approach. In this case, as the objective is to optimize a set of continuous parameters, we are using normal distributions. In each iteration, we fit a univariate

Algorithm 1 EDA

1: Initialize parameters including population size N and selection ratio δ
2: $E(\boldsymbol{\theta}^*) \leftarrow$ Cost function to be optimized
3: Randomly generate initial population $\boldsymbol{G_0}$ with size N
4: **for** $l = 1, 2, \ldots$ until stopping criterion is met **do**
5: $\quad \boldsymbol{G}_{l-1} \leftarrow$ Select δs individuals from G_{l-1} considering $E(\boldsymbol{\theta}^*)$
6: $\quad P(G_l) \leftarrow$ Estimate probabilistic model from G_{l-1}
7: $\quad \boldsymbol{G}_l \leftarrow$ Sample N individuals from $P(G_l)$
8: **end for**=0

normal distribution to each of the parameters (line 6) from the best solutions found in the elite selection (line 5). Each solution evaluation means a set of parameters in the *ansatz*, and an expectation value computation. Then, N new individuals are sampled from the probabilistic model, and the EDA iterates until a stopping criterion is met. The expected behaviour of the algorithm is that the mean of each parameter normal distribution will tend to move in the direction of the optimal value that each parameter should achieve and standard deviations will be reduced.

4 RESULTS

In this section we compare the results obtained by the EDA approach with those obtained by other traditional optimizers when optimizing the parameters of the QAOA *ansatz*. All experiments have been codified using Qiskit-0.19.2 [1], and ran in the same device in order to perform honest comparisons. The implemented approach is available in GitHub[1], and will be pull requested to Qiskit repository. To compare the optimizers, we are using a specific instance of the Max-Cut problem using 10 qubits, and all optimizers are set to the same number of maximum iterations (150 iterations).

Figure 3 shows the relation encountered between the number of layers p of the QAOA *ansatz* and the population size (N) of the EDA. The expectation value improves with increasing N regardless of the number of p layers (see (a)). However, this leads to a higher computation time (see (b)). On the other hand, increasing p regardless of the value of N, the expectation value worsens. To keep a constant expectation value, the population size must be increased while increasing p. Analyzing the computation time, note that increasing the population size for low values of p, the increasing of computation time is not as drastic as increasing N for larger values of p. Thus, it seems that the EDA approach can be a good option for those problems in which the number of layers is not high.

Figure 5 shows a comparison of the EDA approach with traditional optimizers when optimizing the $\boldsymbol{\theta}$ parameters of a QAOA for the same Max-Cut problem. The performance of two different EDAs, eda_20 ($N = 20$) and eda_30 ($N = 30$), is shown.

When comparing the EDA approach with the gradient-free optimizers, the EDA achieves better expectation values independently of p. Comparing the EDA with gradient-based optimizers, the EDA achieves better results than the other optimizers when $p < 6$. Considering the difference between eda_20 and eda_30, by increasing the population size the EDA would achieve better results for $p > 6$. Figure 5 shows a critical difference diagram [9] for the expectation

[1]https://github.com/VicentePerezSoloviev/EDA_QAOA

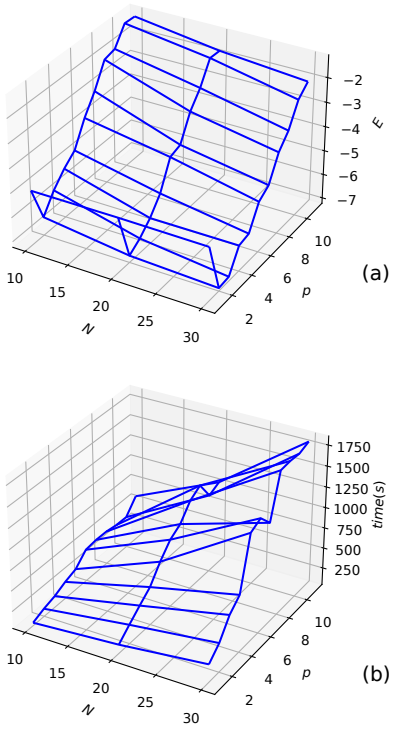

Figure 3: Mean expectation value (a) and mean computational time (b) as a function of the population size (N) and the number of layers (p) for 25 different executions of the EDA optimizer in the QAOA *ansatz*.

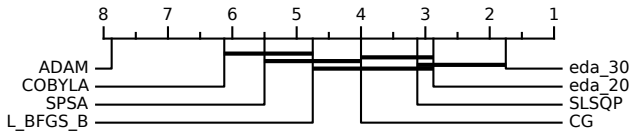

Figure 4: Critical difference diagram [9] by Friedman tests to reject the null hypothesis, and a post-hoc analysis based on the Wilcoxon-Holm method. Horizontal black lines, connect group of algorithms that do not have a significant difference.

value achieved by the optimizers, from where we conclude that EDA achieves statistical improvements. Regarding the computation time, gradient-free optimizers take lower time to be executed than the EDA approach, but at the expense of achieving worse results, and gradient-based optimizers show competitive computation times compared with the EDA approach.

5 CONCLUSIONS

In this work the problem of *ansatz* parameters optimization has been faced by an EDA. Results show the EDA approach compared to some traditional gradient-based and gradient-free optimizers.

Our results show that the EDA is able to converge to better expectation values than any other optimizers for low numbers of layers. When increasing p, the population size of the algorithm should be increased. Analyzing the computation time, the EDA

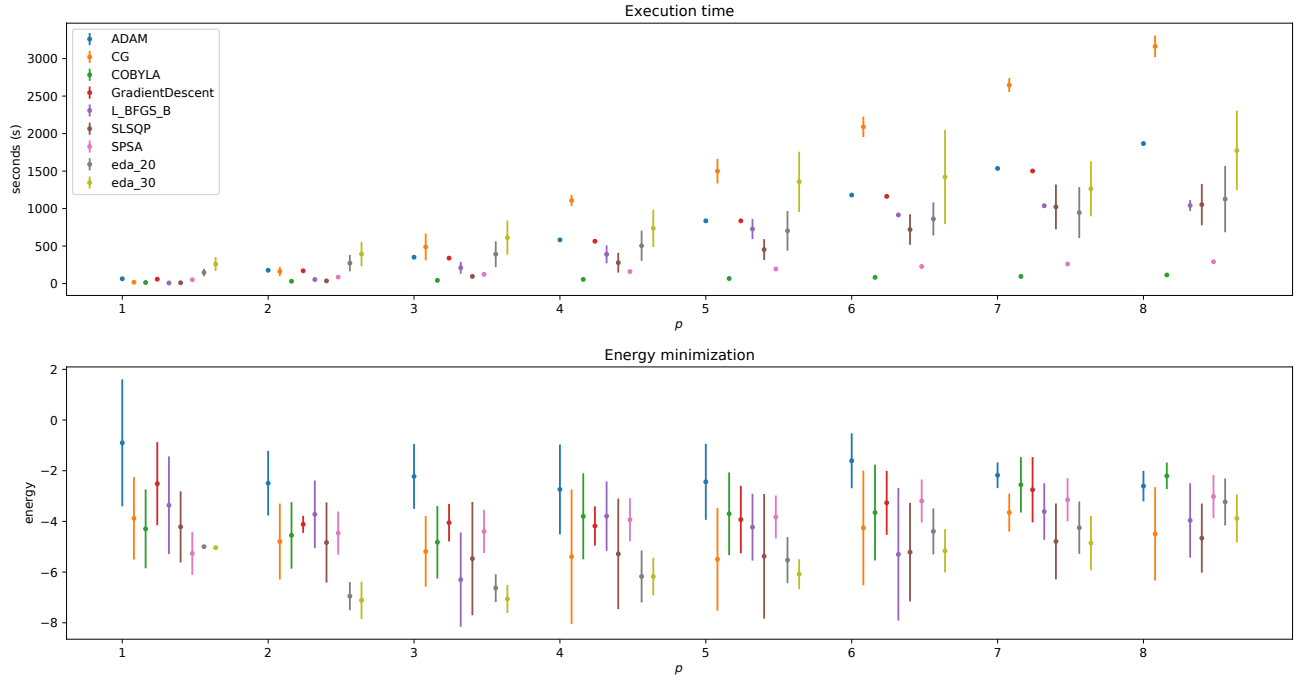

Figure 5: Comparison of EDA approach with other optimizers when optimizing the set of parameters of a QAOA. First panel shows the mean and standard deviation of computational time it takes to return the best solution for different values of p and 25 different executions. Second panel shows the mean and standard deviation of the expectation value the QAOA achieves for different values of p and 25 different executions. Best results after hyper parameter tunning are shown.

approach is more expensive than the gradient-free approaches, but cheaper than other gradient-based algorithms. EDA approach can be a good solution for those situations in which it is more important to achieve a good expectation value than a quick execution of the QAOA. As future steps, it would be interesting to apply more complex probabilistic models into the EDA approach to consider relations between the parameters.

ACKNOWLEDGMENTS

We would like to thank CITIUS and CESGA for access to the computers to run the experiments and IBM Quantum Researchers Program for access to advanced services. This work has been partially supported by the Spanish Ministry of Science and Innovation through the PID2019-109247GB-I00 and the FPI PRE2020-094828 PhD grant of Vicente P. Soloviev.

REFERENCES

[1] Gadi Aleksandrowicz, Thomas Alexander, Panagiotis Barkoutsos, et al. 2021. Qiskit: An Open-source Framework for Quantum Computing. https://doi.org/10.5281/zenodo.2573505
[2] Zahra Beheshti and Siti Mariyam Hj Shamsuddin. 2013. A review of population-based meta-heuristic algorithms. *Int. J. Adv. Soft Comput. Appl* 5, 1 (2013), 1–35.
[3] Kishor Bharti, Alba Cervera-Lierta, Thi Ha Kyaw, et al. 2022. Noisy intermediate-scale quantum algorithms. *Reviews of Modern Physics* 94, 1 (2022), 015004.
[4] Paul T Boggs and Jon W Tolle. 1995. Sequential quadratic programming. *Acta numerica* 4 (1995), 1–51.
[5] Xavier Bonet-Monroig, Hao Wang, Diederick Vermetten, et al. 2021. Performance comparison of optimization methods on variational quantum algorithms. *arXiv preprint arXiv:2111.13454* (2021).
[6] Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. 1995. A limited memory algorithm for bound constrained optimization. *SIAM Journal on scientific computing* 16, 5 (1995), 1190–1208.
[7] Marco Cerezo, Andrew Arrasmith, Ryan Babbush, et al. 2021. Variational quantum algorithms. *Nature Reviews Physics* 3, 9 (2021), 625–644.
[8] Dipankar Dasgupta and Zbigniew Michalewicz. 2014. *Evolutionary Algorithms in Engineering Applications*. Springer.
[9] Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research* 7 (2006), 1–30.
[10] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. 2014. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028* (2014).
[11] Magnus R. Hestenes and Eduard Stiefel. 1952. Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Standards* 49 (1952), 409–435.
[12] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[13] Pedro Larrañaga and Jose A Lozano. 2001. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Vol. 2. Springer.
[14] Andrew Lucas. 2014. Ising formulations of many NP problems. *Frontiers in Physics* (2014), 5.
[15] Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven. 2018. Barren plateaus in quantum neural network training landscapes. *Nature Communications* 9, 1 (2018), 1–6.
[16] Michael JD Powell. 1994. A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in Optimization and Numerical Analysis*. Springer, 51–67.
[17] Sebastian Ruder. 2016. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747* (2016).
[18] Ruslan Shaydulin and Yuri Alexeev. 2019. Evaluating quantum approximate optimization algorithm: A case study. In *2019 Tenth International Green and Sustainable Computing Conference*. IEEE, 1–6.
[19] Vicente P Soloviev, Concha Bielza, and Pedro Larrañaga. 2022. Quantum Approximate Optimization Algorithm for Bayesian network structure learning. *arXiv preprint arXiv:2203.02400* (2022).
[20] James C Spall. 1992. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Trans. Automat. Control* 37, 3 (1992), 332–341.
[21] Utkarsh, Bikash K. Behera, and Prasanta K. Panigrahi. 2020. Solving vehicle routing problem using quantum approximate optimization algorithm. *arXiv preprint arXiv:2002.01351* (2020).