Institute of Architecture of Application Systems

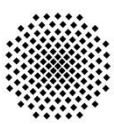
Patterns For Hybrid Quantum Algorithms

Manuela Weigold, Johanna Barzen, Frank Leymann, Daniel Vietz

Institute of Architecture of Application Systems,
University of Stuttgart, Germany
{weigold, barzen, leymann, vietz}@iaas.uni-stuttgart.de

BibTeX:

```
@inproceedings{Weigold2021_HybridPatterns,
  author    = {Weigold, Manuela and Barzen, Johanna and Leymann, Frank and
               Vietz, Daniel},
  title     = {{Patterns for Hybrid Quantum Algorithms}},
  booktitle = {Proceedings of the 15\textsuperscript{th} Symposium and Summer
               School on Service-Oriented Computing (SummerSOC 2021)},
  pages     = {34--51},
  publisher = {Springer International Publishing},
  month     = sep,
  year      = 2021,
  doi       = {10.1007/978-3-030-87568-8\_2}
}
```

University of Stuttgart
Germany

Patterns For Hybrid Quantum Algorithms

Manuela Weigold[0000−0002−4554−260X], Johanna Barzen[0000−0001−8397−7973], Frank Leymann[0000−0002−9123−259X], and Daniel Vietz[0000−0003−1366−5805]

Institute of Architecture of Application Systems,
University of Stuttgart, Universitätsstrasse 38, Stuttgart, Germany
[firstname.lastname]@iaas.uni-stuttgart.de

Abstract. Quantum computers have the potential to solve certain problems faster than classical computers. However, the computations that can be executed on current quantum devices are still limited. Hybrid algorithms split the computational tasks between classical and quantum computers circumventing some of these limitations. Therefore, they are regarded as promising candidates for useful applications in the near future. But especially for novices in quantum computing, it is hard to identify why a particular splitting strategy is proposed by an algorithm. In this work, we describe the best practices for splitting strategies as patterns to foster a common understanding of hybrid algorithms.

Keywords: Quantum Computing · Patterns · Hybrid Algorithms

1 Introduction

Quantum computers are no longer a purely theoretical concept – a first generation of quantum computers is already available to the public [1, 2]. These devices differ considerably from classical computers as their central unit of information is not a classical bit (which can be in a state of either zero or one), but a quantum bit (qubit). Because they are based on qubits, quantum computers are expected to solve certain problems faster than their classical counterparts [3, 4]. Famous examples for quantum algorithms that promise a theoretical linear or exponential speedup are Shor's algorithm [5] for factoring prime numbers or the HHL algorithm [6] for solving linear equations.

However, as current devices still have severe hardware limitations, they are also referred to as *Noisy Intermediate Scale Quantum (NISQ)* devices [7]. Measured by their number of qubits, these devices are of intermediate size (they contain up to a few hundred qubits). Their noisy qubits are only stable for a limited amount of time. Due to their short lifespan, the number of operations that can be executed on a NISQ device is limited. As a consequence, the required operations of many quantum algorithms exceed those of NISQ devices, or they can only be executed for small problems. For example, recent experiments were able to demonstrate Shor's algorithm [5] for factoring small numbers such as 15 or 21 [8]. Given the current state of hardware, Shor's algorithm is one of many algorithms which will not be of much practical use in the near future.

Nevertheless, even with these limitations, first practical applications of quantum computers are expected [7]. Since the size of quantum computations is limited, it has been suggested to use classical computers to overcome the limitations of NISQ devices [9]. Thus, hybrid algorithms strategically split the computational tasks between classical and quantum computers. In the quantum computing literature, an abundance of hybrid algorithms have been proposed and new algorithms appear frequently. However, especially for novices in the field it is often not clear what hybrid algorithm have in common or why they split the computational tasks in a particular manner. This raises the following questions: What are common splitting strategies of hybrid algorithms? What are the benefits and drawbacks of a splitting strategy, and in which contexts are they useful?

To answer these questions, we identify best practices for a *quantum-classical split* and describe them as patterns. In many domains, patterns are an established concept for the documentation of proven solutions for frequently reoccurring problems. Each of our patterns captures the abstract idea behind various quantum algorithms and its relations to other patterns. They are targeted at readers that are already familiar with the basics of quantum computing, such as qubits or quantum circuits. Together, the patterns form a common knowledge base through which different approaches (i) can be understood, and (ii) can be applied in combination to solve a broader problem.

The remainder of this work is structured as follows: we first give an overview of patterns for quantum algorithms of previous works [10, 11] and describe fundamentals of hybrid algorithms in Section 2. Then, we present the six new patterns for hybrid algorithms in Section 3. Related work is described in Section 4. Finally, a conclusion and overview of future work is given in Section 5.

2 Patterns for Quantum Algorithms

In this section, we introduce fundamentals and patterns for quantum algorithms. First, the structure of hybrid algorithms is described and an overview of existing [12, 10, 11] and new patterns is given (Section 2.1). This is followed by Section 2.2 that introduces our pattern format and method for pattern authoring.

2.1 Overview of Patterns for Quantum Algorithms

Figure 1 illustrates the basic structure of hybrid algorithms. The first step on a classical computer is *pre-processing*. A simple example for a pre-processing task is the normalization of input data, however, more complex tasks are also possible. On a quantum computer, the first step is always to *prepare an initial state* of a qubit register. Note that this step can also be used to load data which is usually done by encoding it into the initial state. The resulting state can also be entangled which is one characteristic of quantum states described by patterns of the quantum states category. Additionally, data encoding patterns in the upper left of the figure describe in more detail how this can be realized for a particular

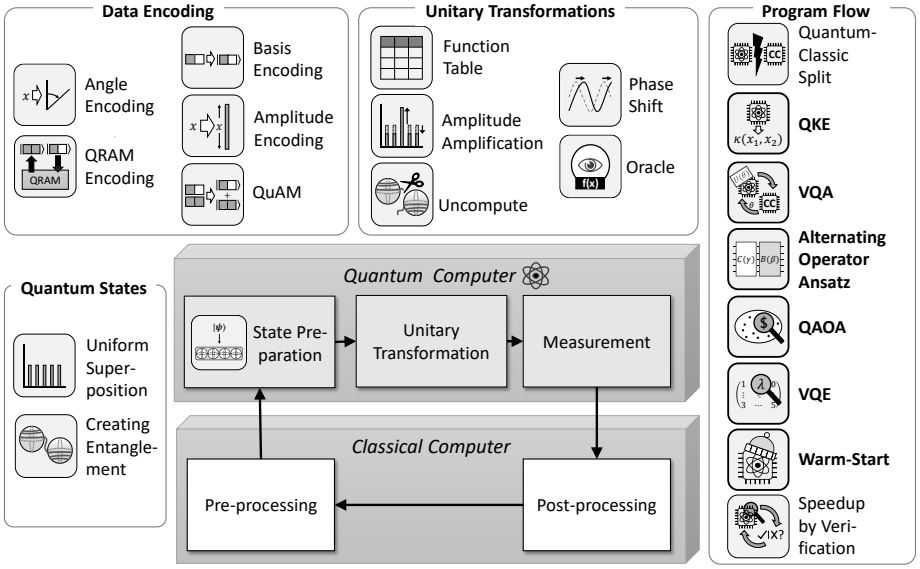


Fig. 1. Overview of previous and new patterns (in bold), adapted from [10].

encoding. In the next step, the quantum computer performs *unitary transformations*. Patterns of the same category describe common transformations, for example, the application of a Phase Shift to mark a particular amplitude. Finally, all or a subset of the qubits are *measured*. The measurement results are *post-processed* on a classical computer. Depending on the overall goal, the algorithm terminates or iterates. Patterns of the category program flow (including the new patterns) describe higher-level strategies for quantum algorithms.

2.2 Pattern Format and Method

For the structure of our patterns, we used a pattern format of previous work [10, 11]. A pattern is introduced by a *Name* and represented graphically as an *Icon*. A pattern name can also be inspired by the name of a prominent algorithm that applies the pattern. Other names under which this pattern may be known are listed as an *Alias*. This is followed by the *Context* of a pattern, which includes a description of the problem. Next, its *Forces* are described that must be considered when solving the problem. Then, the *Solution* is described and often visualized by a *Sketch*. The consequences are described as the *Result* of the solutions. Finally, *Related Patterns* and *Known Uses* are listed.

To identify the patterns, we analyzed the structure of hybrid algorithms documented in the quantum computing literature. If we found various occurrences of a splitting approach (for example, in publications introducing quantum algorithms or in implementations of algorithms) which is also regarded as promising in the literature, we authored a pattern. As quantum computing hardware is still

in an early stage, we do not require concrete implementations as known uses of the patterns. Instead, we focus on finding and describing re-occurring solutions in quantum algorithms.

3 Patterns For Hybrid Algorithms

In this section, we present patterns for hybrid algorithms. In previous work [12], we already introduced the Quantum-Classic Split pattern which motivates that the computational workload is split between quantum and classical resources. As illustrated in Figure 2, the new patterns further refine this pattern by introducing best practices for realizing a Quantum-Classic Split. Thus, the problem, context, and solution of each refining pattern are further specialized towards a concrete use case [13]. As hybrid algorithms are often not invented from scratch but extend previously known splitting strategies, we identified several other refinement relations as shown in Figure 2. In the following, the most abstract Quantum-Classic Split [12] pattern is introduced first via a brief summary and a list of its forces. Note that these are also the forces of all refining patterns. We then present the new patterns, starting with Quantum Kernel Estimator (Section 3.2). This is followed by Variational Quantum Algorithm (Section 3.3) and its further refining patterns (Sections 3.4 to 3.6). Finally, the Warm Start (Section 3.7) pattern is presented. An excerpt of the patterns can be found at http://quantumcomputingpatterns.org.

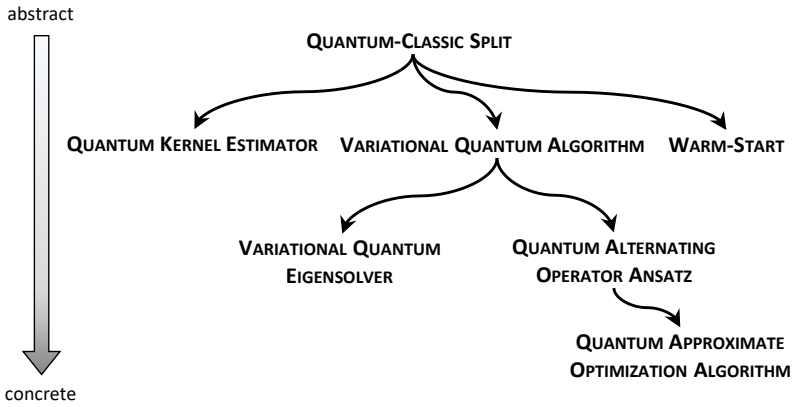


Fig. 2. Overview of all patterns presented in this section. A black arrow indicates that a pattern is further refined by another pattern.

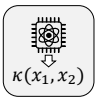
3.1 Quantum-Classic Split

Summary: Solve a given problem using classical as well as quantum resources (also referred to as *hybrid approach*). Depending on the problem, the computational tasks are split between quantum and classical resources.

Forces: A good split of the computational tasks balances the following forces:

(i) Quantum computations are limited: NISQ devices contain a limited number of qubits that are not fully connected. The number of operations which can be executed within the decoherence time of the qubits is also limited [7]. Since the devices are not fully connected, additional operations may be needed to realize operations on multiple qubits [14].
(ii) It is often not possible to divide quantum computations into smaller parts as for example a superposition of inputs needs to be processed at once.

3.2 Quantum Kernel Estimator (QKE)



Use the quantum computer as a kernel estimator for a classical SVM

Context: A classification task must be solved by a support vector machine: given the set of training data $\{x_i\} \subseteq \mathcal{R}^d$ and their corresponding labels y_i, a set of hyper-planes must be found that separates data points according to their labels. Additionally, the distance from the data points to the separating hyper-plane (the *margin*) must be maximized to classify unseen data points with a high probability. However, the data set may not be linearly separable in the original data space.

Solution: Use a quantum computer to compute the kernel function, i.e., the squared inner products between data points in the feature space. Only the value of the kernel function for every pair of data points is needed to optimize the classical SVM. Instead of computing a classical kernel function, a quantum computer uses a quantum feature map ϕ (see Figure 3) to encode a data point x into the Hilbert Space of a quantum system. By encoding a pair of data-points (x, x'), the quantum computer can then estimate the inner product $\langle\phi(x)|\phi(x')\rangle$, for example by using the SWAP test routine [15]. Based on the inner product, the kernel function $K(x, x') = |\langle\phi(x)|\phi(x')\rangle|^2$ can be computed for the data pair and used for the training of the classical SVM.

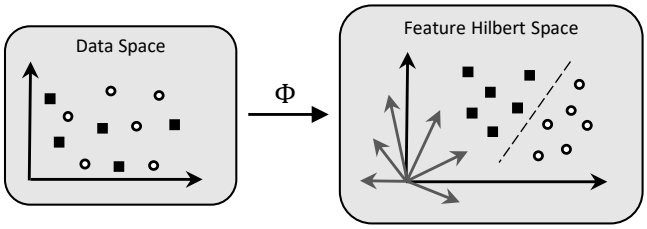


Fig. 3. The quantum feature map ϕ encodes data-points into the Hilbert Space.

Result: The training and classification of the classical support vector machine is efficient if the inner products between data points in the feature space can be evaluated efficiently. Therefore, this setup allows to use feature mappings that cannot be computed efficiently on a classical computer. However, it remains an open research question how a suitable quantum feature map can be chosen.

Related Patterns: This pattern refines Quantum-Classic Split.

Known Uses: In the original paper of the algorithm [16], the authors also demonstrated it in a proof-of-principle experiment on two qubits. Independently, Schuld et al. [15] published the same algorithm and additionally introduced a feature map for continuous-variable quantum systems. Since then, an additional proof-of-principle experiment followed which demonstrated that photonic qubits can be used for this particular task [17]. The approach was also extended in [18]. We implemented this pattern in the QHAna[1] project [19], and also provided a user interface for the selection of different feature maps.

3.3 Variational Quantum Algorithm (VQA)

Optimize the parameters of a quantum circuit on a classical computer

Context: For a given problem, the solution space and, thus, all potential solutions are known. The overall goal is to find the best solution or a solution that is sufficiently good for the task at hand. Since each solution can be evaluated by an objective function, different solutions can be compared with respect to their costs. The objective function C must be *faithful*, i.e., the minimum of C corresponds to the best solution [20]. Ideally, it is also *operationally meaningful* such that smaller values indicate better solutions [20]. However, as the solution space grows exponentially with the problem size, it is difficult to identify the best solution. Consequently, a brute force approach, i.e., calculating the objective

[1] https://github.com/UST-QuAntiL/qhana

function of all solutions, is often only feasible for small problem sizes. Therefore, another approach is to approximate the best solution, i.e., find a solution for which the objective function is close to the minimum.

Solution: To evaluate the cost of a solution, a quantum and a classical computer are used (see Figure 4). The quantum computation (indicated by the quantum circuit in the upper part of Figure 4) is structured as follows:

First, an initial state $|\psi_{in}(x)\rangle$ is prepared which may also depend on a set of input data x. Then, an ansatz circuit $U(\theta)$ is applied to the initial state producing the output state $|\psi_{out}(x,\theta)\rangle$. An ansatz is a unitary operation that is varied by a set of parameters θ. An often-used example of an ansatz is a rotation operation on multiple qubits where the angle depends on θ.

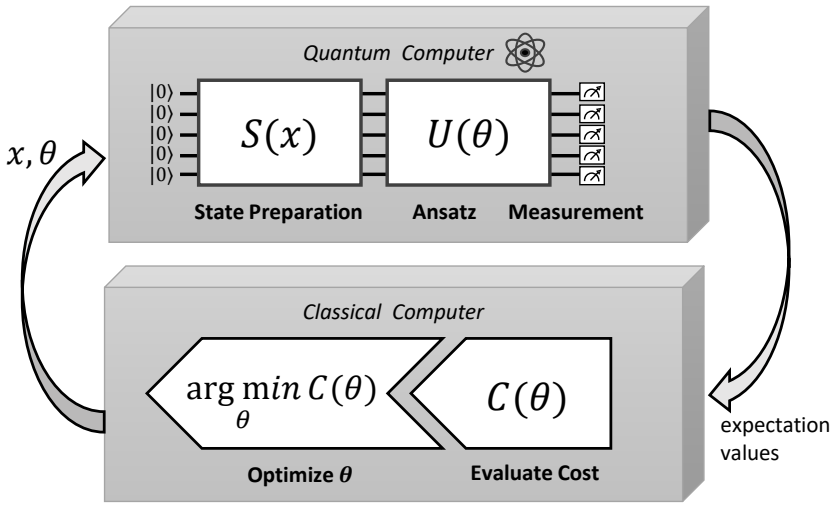


Fig. 4. Quantum-classical setup in which the quantum computation depends on a set of parameters θ updated by an optimization routine running on a classical computer.

Based on the measured expectation values, the value of the objective function for the chosen parameter θ is computed by the classical computer:

$$C(\theta) = \sum_i f_i(\langle O_i \rangle_{|\psi_{out}(x,\theta)\rangle}) \tag{1}$$

where O_i is the observable associated with the i-th measurement, $\langle O_i \rangle_{|\psi_{out}(x,\theta)\rangle}$ is the expectation value for this measurement, and f is a function that associates the expectation values with an overall cost. If the termination condition is fulfilled, e.g., if $C(\theta)$ is sufficiently low, the algorithm terminates. Otherwise, an updated parameter can be obtained by either an optimization technique [20] for a new iteration.

Result: A major drawback of this approach is that it is not trivial to choose a suitable function f, the observables for the measurements, and a good ansatz. How fast the solution converges depends on the objective function as well as the chosen optimization strategy. In particular, *barren plateaus* which are regions of the objective function with a small norm of the gradient can slow down this process even further [20] or even worse may result in non-convergence of the overall algorithm. Since the classical computer performs the optimization, classical computations can also contribute significantly to the overall runtime complexity.

Related Patterns: This pattern refines Quantum-Classic Split [12] and uses State Preparation [12].

Known Uses: There are plenty of known uses for this pattern. For example, variational quantum algorithms for quantum classifiers [21], quantum neural networks [22], a quantum support vector machine [16, 15], and an alternative to Shor's algorithm for factoring prime numbers [23] exist.

3.4 Variational Quantum Eigensolver (VQE)

Approximate the lowest eigenvalue of a matrix

Alias: Quantum Variational Eigensolver (QVE) [21]

Context: The goal is to find the lowest eigenvalue λ of a hermitian matrix H. The hardware requirements for Quantum phase estimation (QPE) are beyond those of NISQ devices, and thus, an approach for NISQ devices is needed.

Solution: Write H as a linear combination of Pauli strings:

$$H = \sum_{\alpha} h_\alpha P_\alpha \tag{2}$$

Following the structure of Variational Quantum Algorithm, prepare trial states by using a suitable ansatz. For example, the unitary coupled cluster ansatz [24] or a hardware-efficient ansatz [20] can be used. Then, the objective function is the sum of the expectation values of the Pauli strings:

$$C(\theta) = \langle\psi(\theta)|H|\psi(\theta)\rangle = \sum_{\alpha} h_\alpha \langle\psi(\theta)|P_\alpha|\psi(\theta)\rangle \tag{3}$$

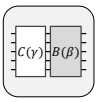
By the variational principle, the expectation value is always greater or equal to the smallest eigenvalue of H. Optimize and update the parameters as described in Variational Quantum Algorithm.

Result: Analogously to Variational Quantum Algorithm, the result, and performance of this approach depend on the objective function, the choice of the ansatz as well as the optimization strategy. The lowest eigenvalue is approximated and can be used to find other eigenvalues (see known uses). This is especially useful for dimension reduction based on principal component analysis (PCA) which requires all eigenvalues of the matrix.

Related Patterns: This pattern refines Variational Quantum Algorithm.

Known Uses: Based on the original algorithm [25] of Peruzzo et al., an approach for finding all eigenvalues of a given matrix was proposed [26]. In quantum chemistry, this algorithm is especially useful for finding ground states of chemical systems and has been applied to various systems [27]. The recently founded Quantum Technology and Application Consortium (QUTAC)[2] identified this algorithm as a possible solution for various industry applications [28].

3.5 Alternating Operator Ansatz (AOA)



Approximate the solution of an optimization problem

Context: A given combinatorial optimization problem must be solved that consists of n binary variables and m clauses which each depend on a subset of the variables. A solution is specified as a bit string $z = z_1 \dots z_n$ that assigns each binary variable z_i to either 0 or 1. The domain is the set of all feasible solutions. Typically a larger configuration space (e.g., all possible bit strings z of length n) is constrained to a subset [29]. The objective function $C(z)$ for a solution is the number of clauses fulfilled by its bit string z:

$$C(z) = \sum_{j=1}^{m} C_j(z) \text{ where } C_\alpha(z) = \begin{cases} 1, & \text{if } C_\alpha \text{ is fulfilled by z} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

Evaluating the objective function for a certain solution is not computationally intensive, but identifying the best solution, i.e., finding the solution which minimizes or maximizes the objective function, is. To circumvent this difficulty, heuristic approaches can be used in order to find a solution whose value of the objective function is close to the maximum or minimum of the objective function.

Solution: Like in VQA, an initial state $|s\rangle$ is created (see Figure 5) and an ansatz is constructed based on a *phase-separating operator* $U(C, \gamma)$ and a *mixing operator* $U(B, \beta)$ which depend on the parameter sets γ and β.

[2] https://www.qutac.de/

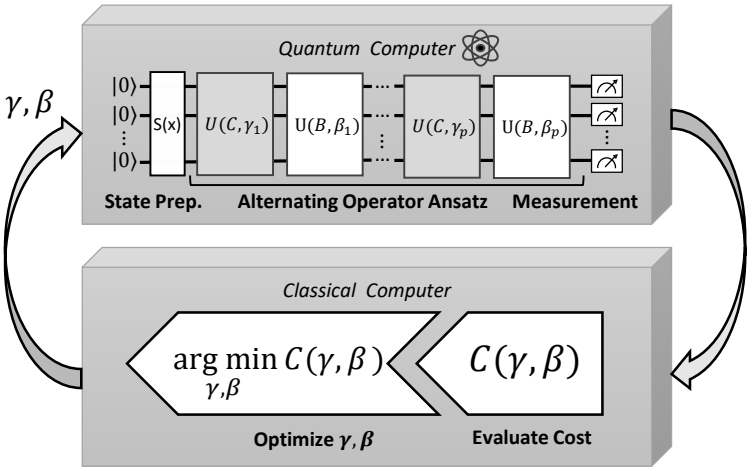


Fig. 5. Quantum Alternating Operator Ansatz.

The initial state $|s\rangle$ is assumed to be created in constant depth, which in general is not possible for some quantum states. For example, $|s\rangle$ can be a state that represents one single solution. Alternatively, a superposition of suitable solutions can be created.

The separating phase operator encodes the objective function and changes the phase of a computational basis state $|y\rangle$ according to $C(y)$:

$$U(C, \gamma) |y\rangle = f(y) |y\rangle \tag{5}$$

For example, $U(C, \gamma)$ can be defined such that a phase shift is applied to $|y\rangle$ for every fulfilled clause. The mixing operator $U(B, \beta)$ changes the amplitudes of the solutions. In particular, it must be possible to transition from any solution to every other solution, i.e., for every pair $(|x\rangle, |y\rangle)$ of computational basis states within the problem domain there exist some parameter β^* for which $U(B, \beta^*)$ provides a transition between them. Therefore, this operator depends on the structure of the domain.

The solution creates on a quantum computer the initial state $|s\rangle$ and applies unitaries drawn from $C(\gamma)$ and $B(\beta)$ in alternation. This results in the following state:

$$|\gamma, \beta\rangle = U(B, \beta_p)U(C, \gamma_p) \dots U(B, \beta_1)U(C, \gamma_1) |s\rangle \tag{6}$$

where γ, β can be initialized randomly at first and $p \in \mathbb{N}$ is a hyper-parameter. Measurement then results in a single solution z for which $C(z)$ can be evaluated. By sampling, the expectation value for γ, β can be determined. Because $|\gamma, \beta\rangle$ is a linear combination of computational basis states, its expectation value is

always smaller or equal to the objective function for the best solution z':

$$\langle C \rangle_{|\gamma,\beta\rangle} = \langle \gamma, \beta | C | \gamma, \beta \rangle = \left\langle \sum x_z \, |z\rangle \, \middle| \, \sum x_z f(z) \, |z\rangle \right\rangle \tag{7}$$

$$= \sum |x_z|^2 f(z) \leq \sum |x_z|^2 f(z') = f(z') = C_{max} \tag{8}$$

The expectation values of variations of the parameters can then be used to optimize the angles for optimizing the objective function and thus, update γ and β. This is repeated until the termination condition is fulfilled (e.g,, a solution z is found for which $C(z)$ is above a certain threshold).

Result: This is an approach that is suitable for NISQ devices and can be adapted to a specific configuration space. As the width of the circuit depends on p, only small values of p are suitable for NISQ devices. However, it is not at all trivial how to construct suitable separating phase and mixing operators for a concrete problem. Although [29] gives several examples of operators for a variety of problems, further research regarding the design of these operators is needed. Analogously to VQA, the choice of the operators (which define the ansatz), the objective function, and the chosen optimization strategy influence how fast the solution converges.

Related Patterns: This pattern refines VQA (the varied parameters are β and γ) and can be combined with Warm Start. Solutions are represented in Basis Encoding [10]. The phase separating operator uses a Phase Shift [12] to mark suitable solutions based on their objective function.

Variations: Besides using computational basis states to represent solutions, also other encodings such as one-hot encoding are possible.

Known Uses: In the original paper [29] that introduces this algorithm, the authors give plenty of examples of how their approach can be applied to a variety of optimization problems and emphasize that is suitable for a broader range of applications. The approach has been used in [30, 31].

3.6 Quantum Approximate Optimization Algorithm (QAOA)

Approximate the solution of an optimization problem

Context: Similar to AOA, an optimization problem must be solved (see Context of AOA for a detailed description) and thus, an objective function is given. In contrast to AOA, the domain of possible solutions is not constrained, and therefore, every bit string $z = z_1 \dots z_n$ is a possible solution of the problem.

Solution: Figure 6 illustrates the *Quantum Approximate Optimization Algorithm (QAOA)* [32] that follows the structure described in AOA. $|s\rangle$ is prepared

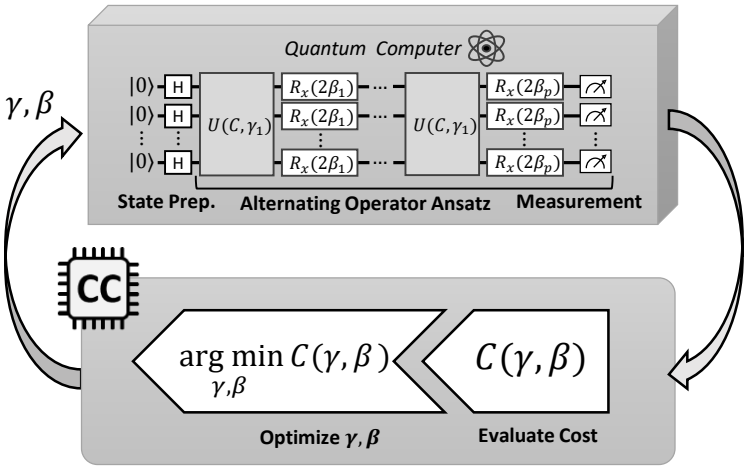


Fig. 6. Overview of QAOA.

as the Uniform Superposition [12] that represents all possible solutions in Basis Encoding [10]. For example, $|0\ldots0\rangle$ represents the solution where all variables are assigned to 0, $|0\ldots01\rangle$ represents $z = 00\ldots01$, etc.

The two operators $U(C,\gamma)$ and $U(B,\beta)$ are defined as follows:

$$U(C,\gamma) = e^{i\gamma C} = \prod_{\alpha=1}^{m} e^{-i\gamma C_\alpha};\ U(B,\beta) = e^{-i\beta B} = \prod_{j=1}^{n} e^{-i\beta\sigma_x^j} \tag{9}$$

For every clause, $U(C,\gamma)$ applies a phase shift $e^{-i\gamma}$ to every computational basis state that fulfills it. As this only marks but does not affect the amplitude of the solutions, $U(B,\gamma)$ is needed to "mix" their amplitudes. This is realized as a rotation around the X-axis with an angle of 2β ($R_x(2\beta)$).

Following the structure described in AOA, $|\gamma,\beta\rangle$ is prepared. Measuring all qubits then results in a single solution (the bit string z), for which the objective function can be evaluated. The parameters β and γ are first chosen at random and then iteratively updated until the termination condition is fulfilled.

Result: This approach is regarded as a promising approach to solve optimization problems on NISQ devices since the required circuit depth is shallow (at most $mp + p$ [32]).

For $p \to \infty$ and suitable small values for γ, β, it approximates the best solution.[3] However, the performance of QAOA depends on the optimization strategy for the angles β, γ and the objective function.

Related Patterns: This pattern creates a Uniform Superposition [12] in the State Preparation [12] of the algorithm. Solutions are represented in Basis Encoding. This pattern refines AOA, and thus, also makes use of Phase Shift [12].

Known Uses: In the initial publication [32], the algorithm was applied to MaxCut. The authors also showed a quantum advantage for the *bounded occurrence problem* [34] which led to the discovery of a better classical algorithm [35]. QUTAC[4] expects that numerous industrial optimization use cases may benefit from quantum computing in the future for which QAOA is one possible approach [28].

3.7 Warm Start

Fine-tune an optimization algorithm by warm starting it

Context: A solution to an optimization problem (see AOA for a detailed description) must be found. Assuming that the *Unique Game Conjecture* (UGC) is true, there is a theoretical upper bound for the approximation ratio of efficient classical methods, i.e., they can only guarantee to approximate the best solution to a certain degree. Since the UGC does not hold for algorithms that exploit entanglement, corresponding quantum algorithms can approximate beyond these bounds.

Solution: Use a classical method to find an approximate solution of the problem. Often, an approximated solution to the original problem can be found solving a simpler, related problem, e.g., for which some of the constraints of the original problem are relaxed.

Result: Warm starting results in a good initial starting solution (near the optimum) which is then used as the start-point for the optimization of the original problem. However, the classical optimization for the related problem also contributes to the overall runtime complexity.

[3] [33] proves this by showing how this algorithm relates to the adiabatic algorithm [33], which also inspired the ansatz of the algorithm.

[4] https://www.qutac.de/

Related Patterns: This pattern can be specialized for QAOA or VQE.

Known Uses: For the MaxCut [36] problem and quadratic optimization problems [37] the warm start of a QAOA was proposed and improved the performance of the optimization. The approach followed in [38] can also be seen as an example for warm starting VQE [37].

4 Related Work

In this work, we use the concept of patterns as introduced by Alexander et al. [39]. The new patterns presented complement quantum computing patterns of previous work [12, 10, 11]. Out of the 15 previous patterns, 13 patterns focus on quantum computations (see Section 2), whereas all new hybrid patterns comprise quantum as well as classical computations. To our best knowledge, there exist no other patterns for quantum computing that conform to the notion of patterns of Alexander et al. [39].

Since the appearance of the first quantum algorithms, numerous algorithms have followed. The most extensive catalog of quantum algorithms known to us is presented at *Quantum Algorithm Zoo*[5] which summarizes algorithms of three categories. An high-level overview of current quantum machine learning algorithms can be found in [40]. The authors also identified whether Grover's search algorithm [41] (a special case of our Amplitude Amplification pattern [12]) was used as subroutine. Since VQE [25] and QAOA [32] are the most prominent variational algorithms, they have been described on a higher level [19, 42, 43, 40] and a more technical level [44, 45] in various other works. In contrast to this work, none of the works describes these proven solutions as interconnected patterns which build on each other to solve larger problems.

In the literature, the term "hybrid quantum-classical algorithms" is sometimes used as a synonym [46, 45] for variational algorithms neglecting hybrid algorithms that are not variational. In this work, we consider hybrid algorithms to include all algorithms consisting of both quantum and classical computations. This for example also includes the hybrid quantum linear equation algorithm [47] or Shor's algorithm [5]. The challenges listed in the Variational Quantum Algorithm pattern are active research areas, such as choosing a suitable ansatz [46] or optimization strategy. An extensive overview of challenges and applications of variational algorithms can be found in [20].

Lloyd [48] proved the universality of the quantum approximate optimization algorithm, i.e., any unitary operator (thus, any quantum algorithm) can in principle be approximated. In [49], the theoretical connections between variational algorithms (which use the structure in our Quantum Variational Algorithm pattern) and kernel-based methods (for which our QKE pattern is the standard example) are explored. The author concludes that especially in the near future, kernel-based methods are an appealing alternative to variational

[5] https://quantumalgorithmzoo.org/

methods. These theoretical results can be regarded as further evidence why the patterns identified by us are proven solutions.

5 Conclusion and Future Work

Especially in interdisciplinary collaborations for building software systems for quantum algorithms, a common understanding of interactions between quantum and classical computers is needed [50]. This work introduced six patterns for hybrid algorithms which refine the Quantum-Classic Split pattern of our previous work [12]. We also describe the relations between them, which reflect that hybrid algorithms often extend previous approaches and are therefore similar in their structure. All of the newly presented patterns try to limit computations on NISQ devices as much as possible. Note that we do not claim that this list of patterns for hybrid algorithms is exhaustive. For example, it should be further investigated if following a divide-and-conquer approach to split up computational tasks until they are suitable for a NISQ device also qualifies as another hybrid pattern. As the hardware improves it may be possible to divide the computational load differently in the future.

The patterns extend our collection of patterns of previous work [12, 10, 11]. In the future, we plan to collect more patterns and further known uses of our patterns within the PlanQK[6] platform. As quantum computing is still in an early stage and hardware improvements constantly open up new possibilities, the patterns presented and their potential applications should be re-evaluated. Following best practices for pattern writing [51], therefore, we plan to revisit and re-evaluate the quality and validity of patterns within the community.

Acknowledgment

This work was partially funded by the BMWi projects *PlanQK (01MK20005N)* as well as the project SEQUOIA funded by the Baden-Wuerttemberg Ministry of the Economy, Labour and Housing. We would also like to thank Felix Truger and Martin Beisel for various discussions about hybrid algorithms.

References

1. "IBM makes quantum computing available on ibm cloud to accelerate innovation," https://www-03.ibm.com/press/us/en/pressrelease/49661.wss, 2016.
2. R. LaRose, "Overview and Comparison of Gate Level Quantum Software Platforms," *Quantum*, vol. 3, p. 130, Mar. 2019.
3. R. Horodecki, P. Horodecki, M. Horodecki, and K. Horodecki, "Quantum entanglement," *Reviews of modern physics*, vol. 81, no. 2, p. 865, 2009.
4. M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information*. Cambridge and New York: Cambridge University Press, 2010.

[6] http://planqk.de/

5. P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th annual symposium on foundations of computer science*. Ieee, 1994, pp. 124–134.
6. A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," *Physical review letters*, vol. 103, no. 15, p. 150502, 2009.
7. J. Preskill, "Quantum Computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, Aug. 2018.
8. M. Amico, Z. H. Saleem, and M. Kumph, "Experimental study of shor's factoring algorithm using the ibm q experience," *Physical Review A*, vol. 100, no. 1, 2019.
9. A. Perdomo-Ortiz, M. Benedetti, J. Realpe-Gómez, and R. Biswas, "Opportunities and challenges for quantum-assisted machine learning in near-term quantum computers," *Quantum Science and Technology*, vol. 3, no. 3, p. 030502, 2018.
10. M. Weigold, J. Barzen, F. Leymann, and M. Salm, "Data Encoding Patterns For Quantum Computing," in *Proceedings of the 27th Conference on Pattern Languages of Programs*. The Hillside Group, 2021, to appear.
11. ——, "Expanding data encoding patterns for quantum algorithms," in *2021 IEEE 18th International Conference on Software Architecture Companion (ICSA-C)*. IEEE, Mar. 2021, pp. 95–101. [Online]. Available: https://ieeexplore.ieee.org/document/9425837/
12. F. Leymann, "Towards a Pattern Language for Quantum Algorithms," in *First International Workshop, QTOP 2019, Munich, Germany, March 18, 2019, Proceedings*. Springer, Apr. 2019.
13. M. Falkenthal, J. Barzen, U. Breitenbücher, C. Fehling, F. Leymann, A. Hadjakos, F. Hentschel, and H. Schulze, "Leveraging Pattern Application via Pattern Refinement," in *Proceedings of the International Conference on Pursuit of Pattern Languages for Societal Change (PURPLSOC 2015)*. epubli, Jun. 2015.
14. F. Leymann and J. Barzen, "The bitter truth about gate-based quantum algorithms in the NISQ era," *Quantum Science and Technology*, pp. 1–28, Sep. 2020. [Online]. Available: https://doi.org/10.1088/2058-9565/abae7d
15. M. Schuld and N. Killoran, "Quantum machine learning in feature hilbert spaces," *Physical review letters*, vol. 122, no. 4, p. 040504, 2019.
16. V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, "Supervised learning with quantum-enhanced feature spaces," *Nature*, vol. 567, no. 7747, pp. 209–212, 2019.
17. K. Bartkiewicz, C. Gneiting, A. Černoch, K. Jiráková, K. Lemr, and F. Nori, "Experimental kernel-based quantum machine learning in finite feature space," *Scientific reports*, vol. 10, no. 1, p. 12356, 2020.
18. R. Ghobadi, J. S. Oberoi, and E. Zahedinejhad, "The power of one qubit in machine learning," 2019, arXiv preprint arXiv:1905.01390.
19. J. Barzen, "From digital humanities to quantum humanities: Potentials and applications," 2021, arXiv preprint arXiv:2103.11825.
20. M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, "Variational quantum algorithms," 2020, arXiv preprint arXiv:2012.09265v1.
21. K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, "Quantum circuit learning," *Physical Review A*, vol. 98, no. 3, 2018.
22. E. Farhi and H. Neven, "Classification with quantum neural networks on near term processors," 2018, arXiv preprint arXiv:1802.06002.
23. E. R. Anschuetz, J. P. Olson, A. Aspuru-Guzik, and Y. Cao, "Variational quantum factoring," 2018, arXiv preprint arXiv:1808.08927.

24. A. G. Taube and R. J. Bartlett, "New perspectives on unitary coupled-cluster theory," *International Journal of Quantum Chemistry*, vol. 106, no. 15, pp. 3393–3401, 2006. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/qua.21198
25. A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, "A variational eigenvalue solver on a photonic quantum processor," *Nature Communications*, vol. 5, no. 1, p. 4213, 2014.
26. O. Higgott, D. Wang, and S. Brierley, "Variational Quantum Computation of Excited States," *Quantum*, vol. 3, p. 156, Jul. 2019. [Online]. Available: https://doi.org/10.22331/q-2019-07-01-156
27. Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. D. Sawaya, S. Sim, L. Veis, and A. Aspuru-Guzik, "Quantum chemistry in the age of quantum computing," *Chemical Reviews*, vol. 119, no. 19, pp. 10 856–10 915, 2019.
28. "Industry quantum computing applications - qutac application group," https://www.qutac.de/wp-content/uploads/2021/07/QUTAC_Paper.pdf, 2021.
29. S. Hadfield, Z. Wang, B. O'Gorman, E. G. Rieffel, D. Venturelli, and R. Biswas, "From the quantum approximate optimization algorithm to a quantum alternating operator ansatz," *Algorithms*, vol. 12, no. 2, 2019. [Online]. Available: https://www.mdpi.com/1999-4893/12/2/34
30. Z. Wang, N. C. Rubin, J. M. Dominy, and E. G. Rieffel, "xy mixers: Analytical and numerical results for the quantum alternating operator ansatz," *Phys. Rev. A*, vol. 101, p. 012320, Jan 2020. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.101.012320
31. M. Fingerhuth, T. Babej, and C. Ing, "A quantum alternating operator ansatz with hard and soft constraints for lattice protein folding," 2018, arXiv preprint arXiv:1810.13411.
32. E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm." [Online]. Available: http://arxiv.org/pdf/1411.4028v1
33. E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, "Quantum computation by adiabatic evolution," 2000, arXiv preprint arXiv:quant-ph/0001106.
34. E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm applied to a bounded occurrence constraint problem," 2015, arXiv preprint arXiv:1412.6062.
35. B. Barak, A. Moitra, R. O'Donnell, P. Raghavendra, O. Regev, D. Steurer, L. Trevisan, A. Vijayaraghavan, D. Witmer, and J. Wright, "Beating the random assignment on constraint satisfaction problems of bounded degree," *CoRR*, vol. abs/1505.03424, 2015. [Online]. Available: http://arxiv.org/abs/1505.03424
36. R. Tate, M. Farhadi, C. Herold, G. Mohler, and S. Gupta, "Bridging classical and quantum with sdp initialized warm-starts for qaoa," 2020, arXiv preprint arXiv:2010.14021.
37. D. J. Egger, J. Marecek, and S. Woerner, "Warm-starting quantum optimization," 2020, arXiv preprint arXiv:quant-ph/0001106.
38. P. K. Barkoutsos, J. F. Gonthier, I. Sokolov, N. Moll, G. Salis, A. Fuhrer, M. Ganzhorn, D. J. Egger, M. Troyer, A. Mezzacapo, S. Filipp, and I. Tavernelli, "Quantum algorithms for electronic structure calculations: Particle-hole hamiltonian and optimized wave-function expansions," *Physical Review A*, vol. 98, no. 2, 2018.
39. C. Alexander, S. Ishikawa, and M. Silverstein, *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, Aug. 1977.

40. S. B. Ramezani, A. Sommers, H. K. Manchukonda, S. Rahimi, and A. Amirlatifi, "Machine learning algorithms in quantum computing: A survey," in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–8.
41. L. K. Grover, "A fast quantum mechanical algorithm for database search," *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing - STOC '96*, 1996.
42. M. Schuld and F. Petruccione, *Supervised Learning with Quantum Computers*, ser. Quantum Science and Technology. Springer International Publishing, 2018.
43. National Academies of Sciences, Engineering and Medicine, *Quantum Computing: Progress and Prospects*. Washington, DC: The National Academies Press, 2019.
44. N. Moll, P. Barkoutsos, L. S. Bishop, J. M. Chow, A. Cross, D. J. Egger, S. Filipp, A. Fuhrer, J. M. Gambetta, M. Ganzhorn, A. Kandala, A. Mezzacapo, P. Müller, W. Riess, G. Salis, J. Smolin, I. Tavernelli, and K. Temme, "Quantum optimization using variational algorithms on near-term quantum devices," *Quantum Science and Technology*, vol. 3, no. 3, p. 030503, 2018. [Online]. Available: https://iopscience.iop.org/article/10.1088/2058-9565/aab822/meta
45. J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, "The theory of variational hybrid quantum-classical algorithms," *New Journal of Physics*, vol. 18, no. 2, p. 023023, feb 2016. [Online]. Available: https://doi.org/10.1088/1367-2630/18/2/023023
46. S. Sim, P. D. Johnson, and A. Aspuru-Guzik, "Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum–classical algorithms," *Advanced Quantum Technologies*, vol. 2, no. 12, p. 1900070, 2019.
47. Y. Lee, J. Joo, and S. Lee, "Hybrid quantum linear equation algorithm and its experimental test on ibm quantum experience," *Scientific reports*, vol. 9, no. 1, p. 4778, 2019.
48. S. Lloyd, "Quantum approximate optimization is computationally universal," 2018, arXiv preprint arXiv:1812.11075.
49. M. Schuld, "Quantum machine learning models are kernel methods," 2021, arXiv preprint arXiv:2101.11020.
50. B. Weder, J. Barzen, F. Leymann, M. Salm, and D. Vietz, "The Quantum Software Lifecycle," in *Proceedings of the 1st ACM SIGSOFT International Workshop on Architectures and Paradigms for Engineering Quantum Software (APEQS 2020)*. ACM, Nov. 2020, Workshop, pp. 2–9. [Online]. Available: https://dl.acm.org/doi/10.1145/3412451.3428497
51. C. Fehling, J. Barzen, U. Breitenbücher, and F. Leymann, "A Process for Pattern Identification, Authoring, and Application," in *Proceedings of the 19th European Conference on Pattern Languages of Programs (EuroPLoP 2014)*. ACM, Jan. 2014.