Scheduling Problem

Solved with Quantum Approximate Optimization Algorithm (QAOA)

Teaser: QAOA is a hybrid near-term quantum algorithm utilizing a problem Hamiltonian and a mixing Hamiltonian to solve combinatorial optimization problems. Its circuits are sparse and relatively shallow, which are both requirements for the execution on NISQ-Devices. The term 'hybrid' refers here to the fact, that part of the algorithm is executed on a quantum computer and another part is executed on a classical computer.

Introduction

Scheduling problems are a branch of combinatorial optimization problems. Thus, the objective is to optimize a schedule in terms of a target value. These schedules could be anything from landing sequences of aircrafts (a.k.a. aircraft landing problem) or sequences of work packages (a.k.a. job shop scheduling problem) to sequences of cars in a paint shop (e.g. binary paint shop problem). The target value would be in these cases e.g. landing the planes within their arrival time window, while maintaining safety distances, reducing the resources, or reducing the number of color swaps for a paint shop. While these problems are already formulated as mathematical models, in general this is not the case. Thus, the first step is to mathematically model the real-world problem. Once such a model is formulated it could be translated in different models. A model, which is suited for combinatorial optimization problems on universal gate-model quantum computers is the Ising-Hamiltonian and can be solved with the Quantum Approximate Optimization Algorithm (QAOA).

Quantum Approximate Optimization Algorithm

QAOA ("Quantum Approximate Optimization Algorithm" or "Quantum Alternating Operator Ansatz") is a modern hybrid algorithm for the universal gate-model quantum computer. Its circuits are sparse and relatively shallow, which are both requirements for the execution on so-called NISQ-Devices (noisy intermediate-scale quantum computers) [1]. The term 'hybrid' refers here to the fact, that part of the algorithm is executed on a quantum computer and another part is executed on a classical computer. In fact, the circuits are parameterized and executed on the quantum computer. The classical computer is utilized to optimize the parameters using an optimization algorithm, which is either simplex or gradient based. Both parts are executed alternating until either optimal parameters are found, or other termination criteria are met.

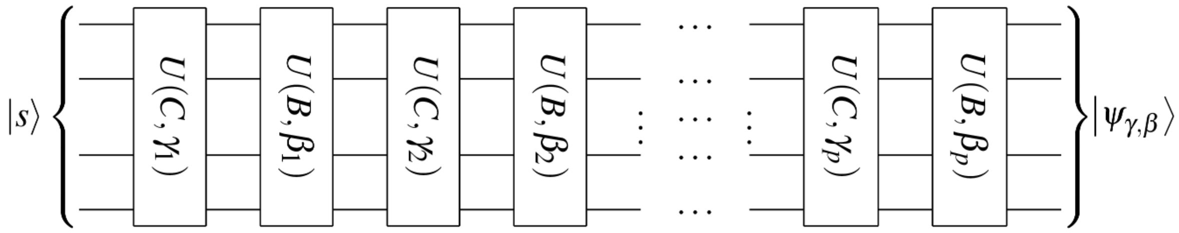

Figure 1: Quantum circuit diagram with QAOA Hamiltonians for p layers.

The circuit itself starts in the equal superposition and is constructed of a problem Hamiltonian U_C (e.g. the Ising-Hamiltonian or a spin glass Hamiltonian) and a mixing Hamiltonian U_B with a known ground state. Both together form a layer. A QAOA circuit can have p layers as shown in Figure 1. However, each Hamiltonian has its own parameter in each layer (i.e. γ_i and β_i). In general, with increasing number of layers the solution gets better, but this also increases the depth of the circuit, which in turn results on current quantum devices in worse solutions. Thus, a good compromise must be found.

The binary paint shop problem

The binary paint shop problem is a simplification of real-world paint shops in the automotive industry, which Volkswagen has recently shown, that it is solvable with QAOA on a quantum computer [2]. The problem itself assumes, that there is a queue of car configurations which should be painted in one of two colors. Each car configuration exists twice in the queue and both configurations should be painted in opposite colors. The sequence of the cars can't be changed. However, the algorithm can decide, if the first occurrence of a configuration should be colored with color one or color two. The second occurrence of the configuration is then, obviously, colored in the other color. It can be shown that already this simplification of the problem is not only NP-hard, but also APX-hard (there is no efficient algorithm, which can approximate the optimal solution).

Solving the binary paint shop problem with QAOA

To solve the binary paint shop problem with QAOA we have to convert the problem into a Hamiltonian. Here we will use the spin glass Hamiltonian, which is of the following form:

$$\frac{1}{2}\sum_{(i,j)\in E} J_{ij}\sigma_z^{(i)}\sigma_z^{(j)}$$

Each car configuration is only one qubit and i and j go over consecutive cars. If these cars are both either seen for the first time or seen for the second time J_{ij} becomes -1 and otherwise $+1$. The mixing Hamiltonian is, as usual, of the following form:

$$\sum_i \sigma_x^{(i)}$$

The solution shows us then the color for the first occurrence of each car configuration.

Outlook

Volkswagen has shown, that a slight modification of the binary paint shop problem could be used to optimize the application of the primer in one of its car facilities [3]. However, this algorithm was executed on a quantum annealer.

Literature

[1] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, und M. D. Lukin, „Quantum Approximate Optimization Algorithm: Performance, Mechanism, and Implementation on Near-Term Devices“, *Phys. Rev. X*, Bd. 10, Nr. 2, S. 021067, 2020, doi: 10.1103/PhysRevX.10.021067.

[2] M. Streif, S. Yarkoni, A. Skolik, F. Neukart, und M. Leib, „Beating classical heuristics for the binary paint shop problem with the quantum approximate optimization algorithm“, Nov. 2020, [Online]. Verfügbar unter: https://arxiv.org/pdf/2011.03403

[3] D-Wave Systems, *Volkswagen: Paint Shop Optimization with Quantum Annealing*, (Nov. 13, 2020). Zugegriffen: Apr. 15, 2021. [Online Video]. Verfügbar unter: https://www.youtube.com/watch?v=Uenk1SF8NsI

Case Study: Portfolio Optimization using Grover Adaptive Search

Introduction

Grover Adaptive Search (GAS) is an extension of the Grover Search (GS) algorithm. Instead of searching for a specific value, GAS can find the lowest (or highest) value without knowing the exact number before. Further, using a special initial state, namely the Quantum Dictionary, this allows us to find the optimal solution of combinatorial optimization problems using its QUBO representation. In this case study we want to examine Grover Adaptive Search on a specific combinatorial optimization problem, namely the problem of Portfolio Optimization.

Grover Search

Grover Search is an already older algorithm developed by Lov Grover in 1996 [1]. It utilizes Amplitude Amplification to find one or more entries in an unsorted set. The entry is a binary number and represented by the qubits. Thus, requiring one qubit per bit.

In general, the algorithm consists of multiple iterations, where one iteration consists of an oracle and a diffusion operator. Starting in an equal superposition over all entries, the oracle flags the entries of interest by inverting their amplitudes. The diffusion operator reflects then all amplitudes at the mean value of all amplitudes. This results in an overall higher amplitude of the entries of interest and a lower one of the others as shown in Figure 1. Applying the iterations, the amplitudes of the entries of interest are first rising, however there is a turning point, where it flips over resulting in lower amplitudes for the entries of interest. When we are looking only for one entry, we have to apply the $\frac{\pi}{4}\sqrt{N}$ iterations, where N is the number of entries. Compared to the best-known classical algorithm we only need $\mathcal{O}\left(\sqrt{N}\right)$ instead of $\mathcal{O}(N)$ runs.

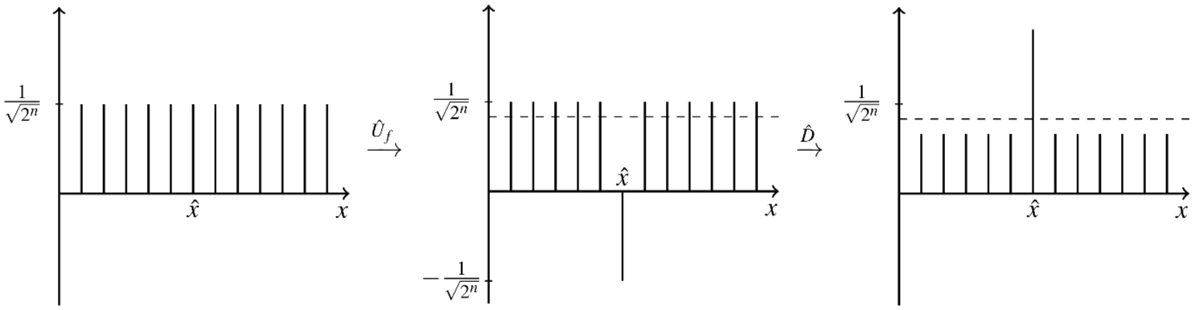

Figure 1: Demonstration of the effect of first applying the oracle and afterwards applying the diffusion operator.

On first sight this algorithm isn't very useful, since we need to know the entries of interest in order to construct the oracle. However, using a Quantum Dictionary [2] we can extend Grover's Algorithm to find a specific solution to a combinatorial optimization problem using QUBOs.

Quantum Dictionaries are maps of keys and values. They use a key and a value qubit register and store all entries in an equal superposition. Most importantly we don't have to enter the value explicitly, but can use a function (e.g., a QUBO), that takes the key and calculates the value. Using Grover Search on the value register, we can now find a key (a combination of a combinatorial problem) for a specific value (a goal function value; the goal function is what we usually want to minimize or maximize). However, this leads us to a new problem since we usually don't know the optimal goal function value in beforehand.

Grover Adaptive Search

Grover Adaptive Search solves the problem of finding a minimal or maximal value in a Quantum Dictionary using Grover Search [3]. For a minimization problem it starts with searching for all values, which are smaller than the highest value which can be represented by the value register or an educated guess. If the Grover Search has found a better value, we use this as our new boundary value. If we can't find a better value after a given number of tries, we assume, that we have found the optimal value and its solution.

Portfolio Optimization

Portfolio Optimization tries to find an optimal subset of an investment universe in terms of risk and expected returns under a given budget.

As input, we have an investment universe of n assets, their expected returns μ (as a vector) and a covariance matrix Σ. Both can be calculated from historic values of the stock market. With a given risk appetite q we can create our object function $\min_{x \in \{0,1\}^n} \left(q x^T \Sigma x - \mu^T x\right)$, where x denotes, if we choose an asset or not. Usually, we have also a budget B, which limits our ability to buy stocks. The budget is added to our object function as a penalty.

A Simple Portfolio Optimization Example

As an example, we use the stocks: GOOG, AAPL, TSLA and JPM, with their historic values from September 2021. We have a risk appetite of $q = 0.5$ and want to buy one quantity of two different stocks. A first glance at the relative historic values of our investment universe (see Figure 2) shows us already the optimal solution 0011 (0 – GOOG, 0 – AAPL, 1 – TSLA, 1 - JPM).

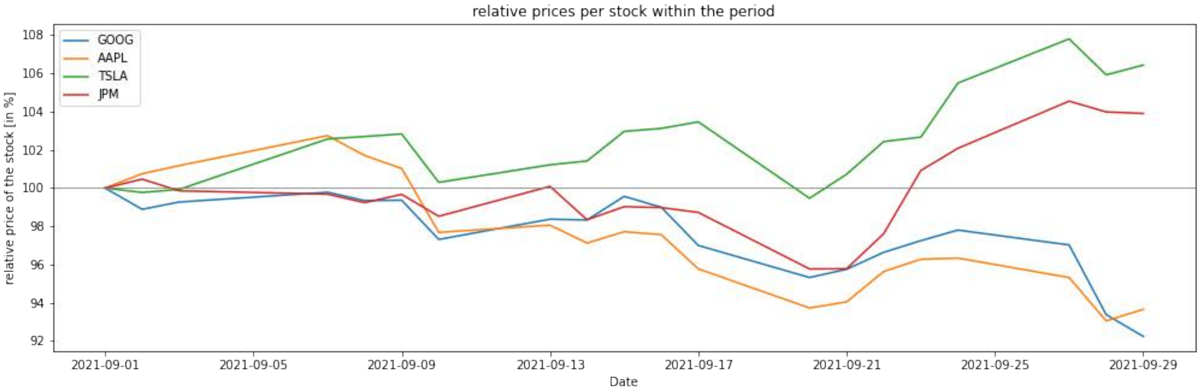

Figure 2: Comparison of the price trend in relative values for the four stocks.

Without going too much into detail of the specific covariance matrix and the returns vector, we have to point out something important: Since we are using bit-values to represent the goal function value, we have to scale the matrix and the vector in such a way, that two goal function values are at least 1 apart from each other. However, this also effects the required number of qubits in the value register (the greater the number the more (qu)bits we need to represent them), thus we have to scale up very carefully.

We have executed 100 experiments for the given problem. Figure 3 shows, that we have a high probability to measure the correct result after executing Grover Adaptive Search. However, we have to admit, that we executed the experiments only on the state vector simulator and not on current hardware. This has two reasons: First, we already need for this small problem 12 fully connected qubits in a quantum computer and second, the current quantum computers are not able to execute quantum circuits with such a high depth (meaning the number of gates executed on one qubit).

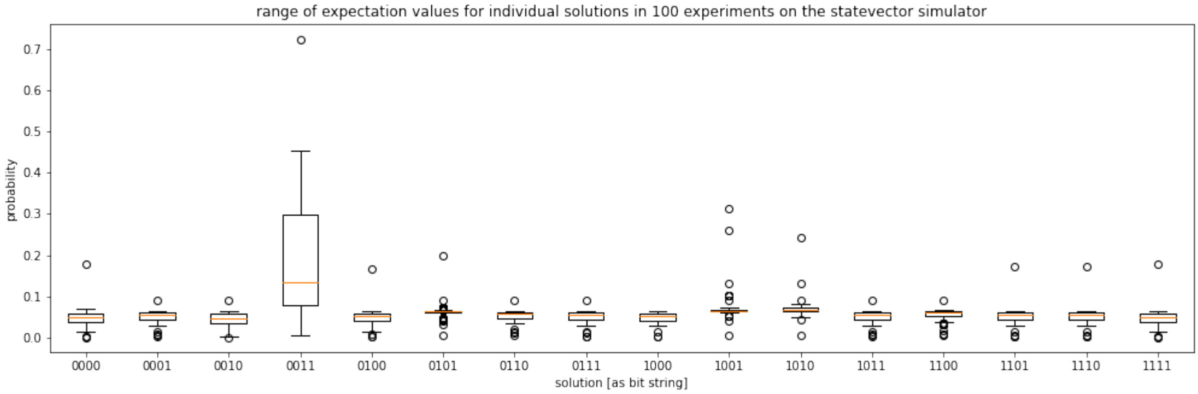

Figure 3: Results of 100 experiments Grover Adaptive Search on the Portfolio Optimization Problem (the optimal solution was 0011 and the budget was two).

Conclusion

In this case study we have shown how to solve the Portfolio Optimization Problem using Grover Adaptive Search. We have seen that Quantum Dictionaries and GAS are not viable on current Quantum Computers. However, we can use the same QUBO to solve the problem using QAOA or VQE. Further experiments have shown, that with a fault tolerant Quantum Computer with sufficient qubits GAS can outperform hybrid algorithms such as QAOA and VQE.

Literature

[1] L. K. Grover, “A fast quantum mechanical algorithm for database search,” *ArXivquant-Ph9605043*, Nov. 1996, Accessed: Oct. 11, 2021. [Online]. Available: http://arxiv.org/abs/quant-ph/9605043

[2] A. Gilliam *et al.*, “Foundational Patterns for Efficient Quantum Computing,” *ArXiv190711513 Quant-Ph*, Jan. 2021, Accessed: Apr. 15, 2021. [Online]. Available: http://arxiv.org/abs/1907.11513

[3] A. Gilliam, S. Woerner, and C. Gonciulea, “Grover Adaptive Search for Constrained Polynomial Binary Optimization,” *ArXiv191204088 Quant-Ph*, Aug. 2020, Accessed: Mar. 01, 2021. [Online]. Available: http://arxiv.org/abs/1912.04088