Wissenschaftliche Arbeit zur Erlangung des Grades
Master of Science
an der Technischen Universität München

Quantum computing approach using QAOA for strategic safety stock placement in multi-echelon inventory with identical service times.

Referent: Prof. Dr. Stefan Minner
Lehrstuhl für Logistik und Supply Chain Management
Technische Universität München

Betreuer: Dr. Chunting Liu

Studiengang: DTU-TUM 1:1 Management MSc Programme

Eingereicht von:
Nicola Antonello
Helene-Mayer-Ring 7 B
80809 München
Tel.: 0039 3455987420
Matrikelnummer: 03801212

An Abstract should not include more than 150 words.

Table of Contents

List of Figures

List of Tables

1 Introduction

The Safety Stock placement problem aims to determine the optimal level of extra inventory that a company should hold to mitigate disruptions while reducing warehousing costs. Safety stocks act as buffers against variations such as unexpected spikes in demand, supplier delays, or transportation disruptions. Maintaining too many safety stocks increases holding costs, while too little can lead to stock shortages, lost sales, and dissatisfied customers.
The main issues concerning this problem are (1) balancing service against cost, (2) dealing with demand and supply uncertainty, and (3) the ineffectiveness of One-Size-Fits-All policies. For simple safety stock placement problems, numerous models and methods have been developed and optimized to identify the ideal safety stock levels. Examples are formula-based approaches, statistical models (service level approaches), and forecast-driven and dynamic techniques.

In supply chain management, several actors come into play, and the objective of the related safety stock placement problem shifts from minimizing the cost of a single inventory to minimizing the overall costs of an entire supply network. However, expanding the problem to multiple actors classifies it as an NP-hard problem, which is recognized as nontrivial to solve with classical computers. This is where Quantum Computing could assist us. QC has emerged as a promising tool for solving problems with enormous search spaces more efficiently than classical methods. Although findings such as Shor's algorithm (Shor, 1996), which has already been improved by Regev (2024), and Grover's algorithm (Grover, 1996) have strengthened this possibility, it remains highly important to understand whether quantum computing could outperform classical computing in every problem, thereby acting as a Universal Turing Machine (Fouché et al., 2024).

Regarding the application of QC to supply chain optimization problems, much research has been done to find appropriate algorithms that are useful for practical uses. Specifically, in combinatorial optimization, a successful application is presented by Xie et al. (2024) who solved a CVRP by combining a Quantum Alternating Operator Ansatz AOA with a Grover Mixer operator. Additionally, another example is shown by Bentley et al. (2024) where they also used QC to solve a CVRP using the Quantum Approximate Optimization Algorithm.
Regarding practical use cases of QC in Inventory Optimization, there is not much work done in this matter. Interesting papers are Sharma and Lau (2024) and Sharma et al. (2024) where two different quantum approaches are used to solve the Newsvendor problem.

In general, Ramya (2024) states three popular quantum algorithms for solving inventory management optimization problems: the Variational Quantum Eigensolver VQE, the Quantum Approximate Optimization Algorithm QAOA and the Quantum Annealing QA. This paper presents a way to solve the non-linear model for placing strategic safety stock in multi-echelon inventory with identical service times described by Grahl et al. (2016) using the QAOA on the IBM-kyiv QPU.
Additionally, the work presented in this paper is partially inspired by the publication of Sharma et al. (2024) in which discrete (non-binary) variables describing the demand in the newsvendor problem are encoded using multiple qubits, this technique is commonly known as integer encoding. To do that, the optimization model is "constrained" to integer variables to make possible an implementation on an actual QPU. Encoding a float variable with a reasonable decimal precision requires many qubits and due to the lack of these in current QPUs, it is not possible to implement them to perform Mixed Integer Programming in a Quantum Computer.

2 The guaranteed service time safety stock allocation problem with identical service times

2.1 The General Model

In this section, the generalized model proposed by Minner (2000) is described. The safety stock placement problem in a supply network can be seen as an acyclic graph G, whose nodes correspond to the stock points of the supply chain. Below is the list of entities that describe the model:

Parameters

A first-stage stock points without predecessors

P intermediate-stage stock points

E final-stage stock points without successors

n stock points of the supply network

i i^{th} stock point of the supply network

$v(i)$ predecessor node of node i

$n(i)$ successor node of node i

h_i inventory holding costs per unit and unit of time at stock point i

SL_i non-stock out probability service level of stock point i

$A(i,j)$ material flow requirements between two stock points i and j

$a_{i,j}$ dependent internal demand input coefficients

λ_i processing time at stock point i assumed to be deterministic and integer multiple of the review period given all required materials. It also includes the review period

$\overline{ST}_e$ service time of node $e \in E$ are given

$W(A,i)$ set of all paths from node set A to node i

$W(A,E)$ set of all paths between the first-stage and the final-stage nodes

$k_i(SL_i) = \Phi_{0,1}^{-1}(SL_i)$ safety factor that depends on the service level SL_i. $\Phi_{0,1}^{-1}$ is the inverse of the cumulative distribution function of the standard normal distribution

w sequence of nodes that forms a path in the supply network

Variables

ST_i service time of node i

T_i coverage time of node i

$X_i = \sigma^2 T_i$ is the variance of the aggregate demand covered by stock point i

Model
objective

$$min \quad C = \sum_{i=1}^{n} h_i k_i(SL_i)\sqrt{X_i} \tag{2.1}$$

subject to

$$\max_{w \in W(A,i)} \left\{ \sum_{j \in w} (\lambda_j - T_j) \right\} \geq 0, \qquad \forall i = 1, 2, \ldots, n \tag{2.2}$$

$$\sum_{j \in w} T_j \geq \sum_{j \in w} \lambda_j - \sum_{i \in \{w \cap E\}} \overline{ST}_i, \qquad \forall w \in W(A, E) \tag{2.3}$$

$$X_i = \sigma_i^2 T_i, \quad \text{with} \quad \sigma_i^2 = \sum_{e \in E(i)} a_{ie}^2 \sigma_e^2 \tag{2.4}$$

$$T_i = \max_{j \in v(i)} \{ST_j\} + \lambda_i - ST_i \tag{2.5}$$

$$T_i, ST_i, X_i \geq 0, \qquad \forall i = 1, 2, \ldots, n \tag{2.6}$$

Constraint (2) ensures that service times cannot be negative. Inequality (3) enforces the network's cumulative coverage of processing times across all paths. Equation (4) describes the relation between the variance of the aggregated demand of a stock point X_i and its coverage time T_i. The equation (5) tells that the coverage time T_i is the replenishment time $\max_{j \in v(i)}\{ST_j\} + \lambda_i$ minus the service time ST_i. Finally, constraints (5) ensure that coverage times T_i, service times ST_i, and the variance of the aggregated demand X_i cannot be negative.

2.2 Case Study Model

The Model that we propose to solve in this paper is a 3-stage serial supply network represented by the following scheme:

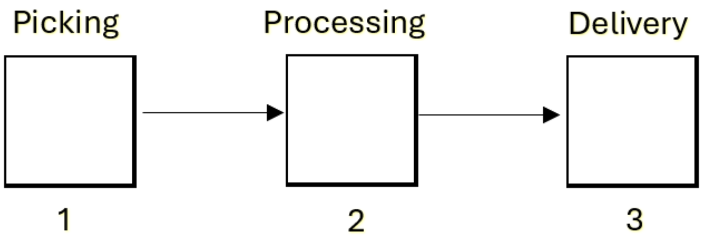

Figure 2.1: 3-stage serial supply network.

For this case, the following parameter values have been chosen:

$$h_1 = h_2 = h_3 = 1$$

$$SL_1 = SL_2 = SL_3 = 1$$

$$\lambda_1 = \lambda_2 = \lambda_3 = 1$$

$$\overline{ST}_e = ST_3 = 1$$

$$\sigma_1 = \sigma_2 = \sigma_3 = 2$$

3 Quantum Approximate Optimization Algorithm QAOA

The Quantum Approximate Optimization Algorithm QAOA is part of a bigger class of algorithms called Variational Quantum Algorithms (VQAs) (Bharti et al., 2021). This class of algorithms is based on the Adiabatic Theorem which states that given a time-dependent Hamiltonian $H(s)$, $s \in [0, 1]$, and a system initialized at time $t = 0$ in the ground state of $H(0)$. Let the system evolve according to the Hamiltonian $H(t/T)$ from time $t = 0$ to time T. We refer to such a process as an adiabatic evolution according to H for time T. The adiabatic theorem affirms that for large enough T the final state of the system is very close to the ground state of $H(1)$ (Aharonov et al., 2008).
An interesting algorithm of this class is the Quantum Adiabatic Algorithm (QAA) which relies on a continuous slow change of the system's Hamiltonian. If the evolution is slow enough, the system stays in its ground state throughout the process, and it will end in the ground state of the final Hamiltonian.
Unfortunately, current quantum devices, also referred to as NISQ (Noisy Intermediate-Scale Quantum) devices, are not able to run effectively such an algorithm due to the limitation of error handling and decoherence time. Here is where the QAOA comes to overcome this limitation. QAOA can be seen as a discrete implementation of the adiabatic evolution performed by the QAA, by applying a series of alternating quantum gates derived from two Hamiltonians: one representing the problem, also referred to as cost Hamiltonian, and one serving as a mixer. This pair of Hamiltonian operations is called step or trotter. Ideally, for an infinite number of trotters, the QAOA coincides with the QAA. Before continuing with the general implementation of the QAOA, it is needed to know that generally, a cost function $C(x)$ can be mapped to a cost Hamiltonian $\hat{H}_C$ such that:

$$\hat{H}_C \left| x \right\rangle = C(x) \left| x \right\rangle , \tag{3.1}$$

where x is the quantum state that encodes the output bitstring.

According to the work of Blekos et al. (2023), the steps to implement the QAOA for a general problem are

1. Firstly, we need to encode the cost function of the problem into a Hamiltonian $\hat{H}_C$ and define the mixing Hamiltonian $\hat{H}_M$. For combinatorial optimization,

a really useful Hamiltonian is the *Ising*-Hamiltonian:

$$\hat{H}_{Ising} = \sum_{i,j} J_{i,j} Z_i Z_j + \sum_i h_i Z_i$$

where Z_i is the Pauli-Z (-X) operator acting on qubit i, $J_{i,j}$ represents the coupling strength between qubits i and j, h_i is the local magnetic field applied to qubit i. Then, we need to define the cost Hamiltonian $\hat{H}_C$. The simplest mixing Hamiltonian possible is

$$\hat{H}_M = \sum_{i=1}^{n} X_i$$

2. Next, it is required to set up the initial quantum state $|s\rangle$ as

$$|s\rangle = |+\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}\in\{0,1\}^n} |\mathbf{x}\rangle\,,$$

where n is the number of qubits of the system.

3. Now, we have to translate the two Hamiltonians into two unitary operations applicable to a quantum circuit, also called the cost and the mixing layers. The two unitaries are

$$\hat{U}_C(\gamma) = e^{-i\gamma\hat{H}_C}$$
$$\hat{U}_M(\beta) = e^{-i\beta\hat{H}_M}$$

where γ and β are variational parameters of the specific circuit.

4. Define the number of trotters (pairs of layers) of the QAOA $p \geq 1$. Then initialize $2p$ variational parameters $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_p)$ and $\beta = (\beta_1, \beta_2, \ldots, \beta_p)$ such that $\gamma_k \in [0, 2\pi)$ and $\beta_k \in [0, \pi)$. The final state output is

$$|\psi_p(\gamma,\beta)\rangle = e^{-i\beta_p\hat{H}_M} e^{-i\gamma_p\hat{H}_C} \ldots e^{-i\beta_1\hat{H}_M} e^{-i\gamma_1\hat{H}_C} |s\rangle$$

5. Now, it is possible to compute the expectation value of the Hamiltonian $\hat{H}_C$

$$F_p(\gamma,\beta) = \langle\psi_p(\gamma,\beta)|\,\hat{H}_C\,|\psi_p(\gamma,\beta)\rangle$$

6. Finally, it is possible to iteratively optimize the choice of the parameters γ and β by using a classical optimization algorithm. The goal would be to find (γ^*, β^*) that maximize the expectation value of $F_P(\gamma,\beta)$

$$(\gamma^*, \beta^*) = \arg\max_{\gamma,\beta} F_p(\gamma,\beta)$$

4 QAOA for guaranteed service time safety stock allocation problem

To align with the form of the Ising-Hamiltonian

$$\hat{H}_{Ising} = \sum_{i,j} J_{i,j} Z_i Z_j + \sum_i h_i Z_i$$

a mathematical framework called QUBO (Quadratic Unconstrained Binary Optimization) has become very popular for such problems. The general formulation of a QUBO problem is given by:

$$\min_{x \in \{0,1\}^n} \quad f(x) = x^T Q x,$$

where:

x is an n-dimensional binary vector (each $x_i \in \{0, 1\}$),

Q is an $n \times n$ real-valued matrix containing the coefficients for both the linear and quadratic terms.

This framework is widely used to express many combinatorial optimization problems in a form that can be easily implemented as quantum circuits. The main reason why it is so useful is because its formulation matches the one required by the Ising-Hamiltonian:

1. The QUBO model is unconstrained. To ensure the feasibility of the solutions with respect to the original problem, the constraints are introduced into the objective function via a Lagrangian reformulation. In this approach, penalty terms are introduced to the objective function to avoid any solution that violates the original constraints.
2. All the variables that come into play are binary, which means that it is possible to represent them using qubits.
3. The objective function is of the 2nd order, in other words, quadratic.

It is really important now to state that, to proceed with the QUBO model, all the decision variables of the original problem are bounded to integer variables. In this way, we can map their values to a simple combination of bits. Take a look at 1 for a detailed look of the calculation process.
The resulting general QUBO formulation for the Strategic Safety Stock placement in multi-echelon inventory with identical service times is

$$
\begin{aligned}
\min C = \sum_{i=1}^{n} h_i \, \Phi_{0,1}^{-1}(SL_i) \; \sigma_i \left(\sum_b 2^b z_{i,b} \right) + \\
+ \alpha \sum_{i=1}^{n} \left(1 - \sum_{k=1}^{|W(A,i)|} y_{k,i} \right)^2 + \\
+ \beta \sum_{i=1}^{n} \sum_{k=1}^{|W(A,i)|} \Bigg(\sum_b 2^b u_{i,b} - \sum_{j \in w_k} \Big(\lambda_j - \Big(\sum_b 2^b z_{j,b} \Big)^2 \Big) \\
- \Big(\sum_b 2^b M_b \Big) \big(1 - y_{k,i} \big) + \sum_b 2^b V_{k,i,b} \Bigg)^2 + \\
+ \gamma \left(\sum_{i=1}^{n} \lambda_i - \sum_{i \in \{w \cap E\}} \sum_b 2^b ST_{i,b} - \sum_{i=1}^{n} \left(\sum_b 2^b z_{i,b} \right)^2 + \sum_b 2^b Q_b \right)^2 + \\
+ \rho \sum_{i=1}^{n} \left(\left(\sum_b 2^b z_{i,b} \right)^2 - \sum_b 2^b H_{i,b} - \lambda_i + \sum_b 2^b ST_{i,b} \right)^2 + \\
+ \eta \sum_{i=1}^{n} \sum_{j \in v(i)} \left(\sum_b 2^b ST_{j,b} - \sum_b 2^b H_{i,b} + \sum_b 2^b R_{i,j,b} \right)^2 + \\
+ \epsilon \sum_{i=1}^{n} \sum_{j \in v(i)} \left(\sum_b 2^b H_{i,b} - \sum_b 2^b ST_{j,b} - \Big(\sum_b 2^b N_b \Big) \big(1 - x_{i,j} \big) + \sum_b 2^b U_{i,j,b} \right)^2 + \\
+ \delta \sum_{i=1}^{n} \left(1 - \sum_{j \in v(i)} x_{i,j} \right)^2 . \qquad (4.1)
\end{aligned}
$$

where

b identify the b^{th} bit of an integer variable.

$z_i = \sqrt{T_i}$

α, β, γ, ρ, η, ϵ and δ are the introduced penalty terms

$y_{k,i}$ and $x_{i,j}$ are auxiliary binary variables

$V_{k,i,b}$, Q_b, $R_{i,j,b}$ and $U_{i,j,b}$ are slack variables used to transform inequality constraints of the original model to equality constraints.

H_i, b and u_i, b are variables used to linearized the original *max* constraints

M and N are "Big-M" notation parameters.

5 Results

The model (23) is implemented in a python jupyter notebook[1]. The IBM-kyiv QPU is used for the experiment, which has an *Eagle r3* quantum processor with up to 127 qubits available.
Here is the topology of the QPU used

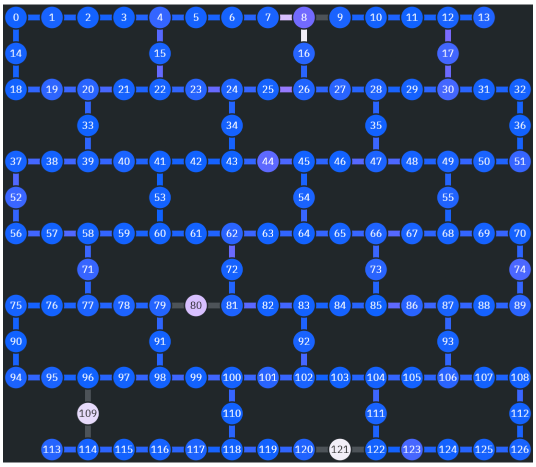

Figure 5.1: IBM-kyiv Eagle r3 qubit topology

The QAOA applied to the case is composed of only two trotter steps. This choice is due to the limitation of the IBM quantum platform limitation usage. This choice led us to a QAOA with just two variational parameters γ_0 and β_0.

The pure quantum circuit representing the QUBO model is the following

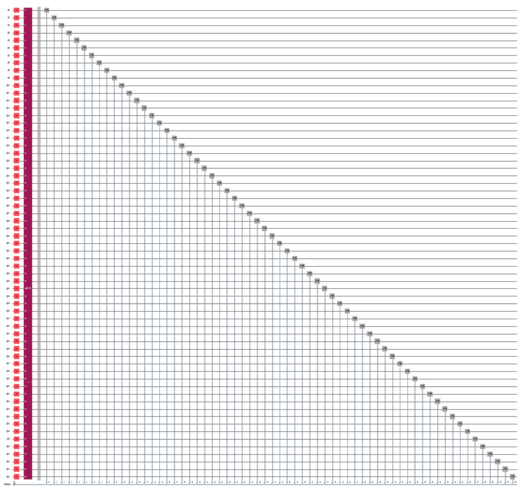

Figure 5.2: Unoptimized quantum circuit

[1]The code used in for this paper is available at https://github.com/NikTheBoss97/QCforSCO.git

To optimize the circuit, in terms of circuit depth, we performed the transpilation of the circuit using the qiskit transpiler. Transpilation is the process of rewriting a given input circuit to match the topology of a specific quantum device, and optimize the circuit instructions for execution on noisy quantum computers. The optimization level set was 3, the highest possible. With this level, the following optimization techniques have been performed:

Optimization level 2 + heuristic optimized on layout/routing further with greater effort/trials

Resynthesis of two-qubit blocks using Cartan's KAK Decomposition.

Unitarity-breaking passes: moves the measurements around to avoid SWAPs and removes gates before measurements that would not affect the measurements

The transpiled circuit obtained is

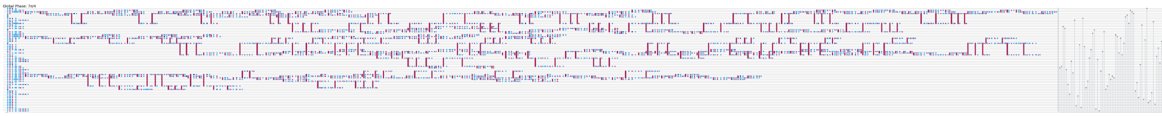

Figure 5.3: Transpiled quantum circuit

Then, we performed a classical optimization of the variational parameters γ_0 and β_0 using the *COBYLA* method.

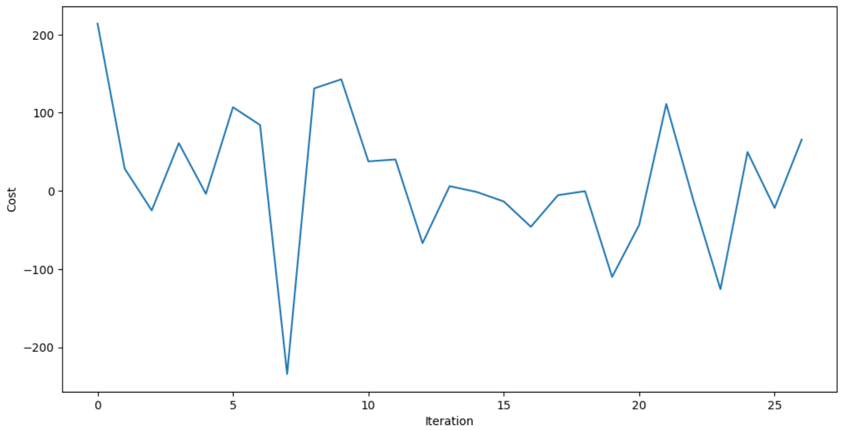

Figure 5.4: Parameter objective function values over iterations

And finally, the optimized quantum circuit has been run in the IBM-kyiv QPU. The results are the following:

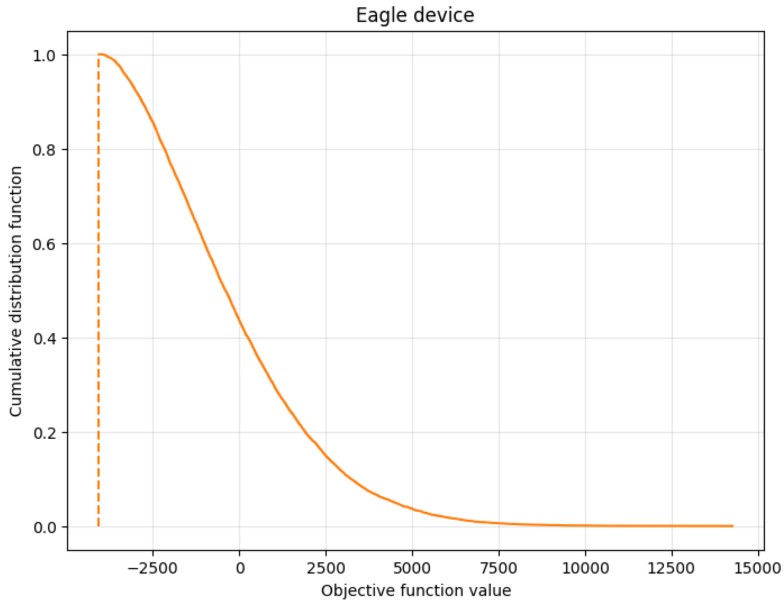

Figure 5.5: Cumulative probability distribution function of the solutions

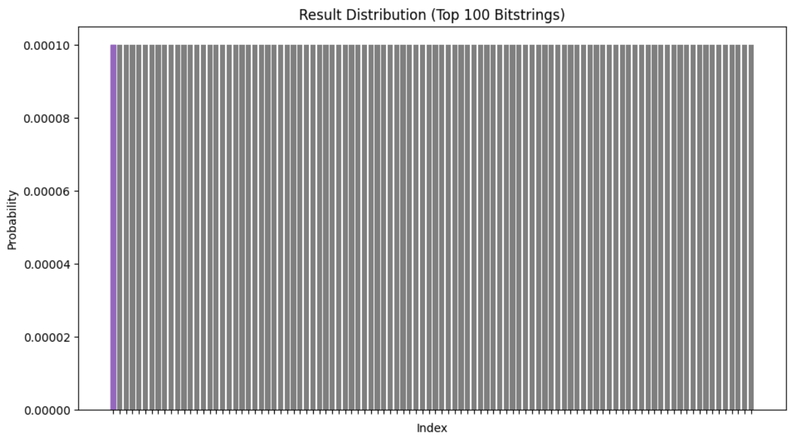

Figure 5.6: Most likely bitstrings and their probabilities

As shown in figure 5.6, the quantum algorithm has failed to identify effectively a single or a reasonable number of possible solutions. However, the circuit still provides the majority of the solutions with an objective value close to the optimal one which is 6.56 as shown in figure 5.5.

6 Evaluation and Conclusion

There are many classes of problems in operations research that are nontrivial to find a solution to. Several efforts were and are still made to these kinds of problems to find ways to solve them faster. Techniques such as machine learning algorithms, meta-heuristics algorithms and big-data approaches have provided us with notable improvements in this matter. Unfortunately, these techniques are extremely problem-specific and, therefore, hard to adapt to new problems. However, in the last decade, quantum computing has obtained popularity thanks to the improvements achieved by different companies such as D-Wave, IBM, Google and Microsoft.

Quantum computing has been recognized with a strong potential in solving optimization problems that are extremely hard or even intractable by classical computers. In this regard, supply chain management could strongly benefit from this technology and much research is ongoing in this direction. Different algorithms have been tested and compared against classical techniques to understand the true capabilities of QC. Algorithms such as Variational Quantum Eigensolver VQE, Quantum Adiabatic Algorithm QAA, and many more have proven their usefulness in practical examples. In particular, one that has become popular due to its flexibility and customization is the Quantum Approximate Optimization Algorithm QAOA, which is a discrete approximate version of the QAA. In particular, the QAOA has been used to solve small combinatorial problems such as the CVRP and the TSP. Unfortunately, there is not much literature when talking about inventory optimization. In this paper, we tried to implement a basic QAOA algorithm to solve a strategic safety stock placement problem in a multi-echelon inventory with identical service times in the specific case of a 3-stage serial supply network.

The algorithm produced is not completely satisfying since the pool of the most probable solutions is very big and the feasibility among these is often not met. This unfortunate outcome can be a consequence of one or multiple of the following issues:

1. Due to usage limitation constraints, the QAOA applied has only a single trotter step. This makes it difficult for the QPU to approximate the behavior expected of the idea Quantum Adiabatic Algorithm.

2. The QAOA itself could not be sufficient to identify a feasible solution for the final QUBO model reliably. This is also stated by Xie et al. (2024) in work on finding an efficacious quantum algorithm for solving the CVRP.

3. The qiskit.circuit.library.QAOAAnsatz tool, part of the qiskit library, imple-

ments a problem-specific ansatz. Probably, it is worth trying to implement different ansatzes and see if the quantum circuit improves.

4. The search space might be reduced by implementing new ways to encode integer variables into set binary variables (Karimi and Ronagh, 2017).
5. The mixer Hamiltonian implemented by qiskit might be not the best one. One could try to see whether different mixer Hamiltonians perform better.

Nonetheless, the results seem to confirm that the direction taken might be correct as the highly probable solutions fell around the optimal value which was 6.54 as shown in figure 5.5.

Bibliography

Aharonov, Dorit, Wim van Dam, Julia Kempe, Zeph Landau, Seth Lloyd, Oded Regev. 2008. Adiabatic quantum computation is equivalent to standard quantum computation. *a*rXiv:quant-ph/0405098v2 Preprint available at `https://arxiv.org/abs/quant-ph/0405098v2`.

Bentley, Christopher D. B., Samuel Marsh, André R. R. Carvalho, Philip Kilby, Michael J. Biercuk. 2024. Quantum computing for transport optimization. *a*rXiv preprint **quant-ph**(2206.07313). URL `https://arxiv.org/abs/2206.07313`.

Bharti, Kishor, Alba Cervera-Lierta, Thi Ha Kyaw, Tobias Haug, Sumner Alperin-Lea, Abhinav Anand, Matthias Degroote, Hermanni Heimonen, Jakob S. Kottmann, Tim Menke, Wai-Keong Mok, Sukin Sim, Leong-Chuan Kwek, Alán Aspuru-Guzik. 2021. Noisy intermediate-scale quantum (nisq) algorithms. *a*rXiv preprint (2101.08448v2). URL `https://arxiv.org/abs/2101.08448`.

Blekos, Kostas, Dean Brand, Andrea Ceschini, Chiao-Hui Chou, Rui-Hao Li, Komal Pandya, Alessandro Summer. 2023. A review on quantum approximate optimization algorithm and its variants. *a*rXiv:2306.09198v2 URL `https://arxiv.org/abs/2306.09198v2`.

Fouché, Willem, Johannes Heidema, Glyn Jones, Petrus H. Potgieter. 2024. Deutsch's universal quantum turing machine (revisited). *a*rXiv preprint **quant-ph**(0701108v1). URL `https://arxiv.org/abs/quant-ph/0701108v1`.

Grahl, Jörn, Stefan Minner, Daniel Dittmar. 2016. Meta-heuristics for placing strategic safety stock in multi-echelon inventory with differentiated service times. *A*nnals of Operations Research **242** 489–504. doi:10.1007/s10479-014-1635-1. URL `https://doi.org/10.1007/s10479-014-1635-1`.

Grover, Lov K. 1996. A fast quantum mechanical algorithm for database search. *P*roceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC). ACM, Philadelphia, PA, USA, 212–219. URL `https://arxiv.org/abs/quant-ph/9605043`.

Karimi, Sahar, Pooya Ronagh. 2017. Practical integer-to-binary mapping for quantum annealers. *a*rXiv preprint **quant-ph**(1706.01945v1). URL `https://arxiv.org/abs/1706.01945`.

Minner, Stefan. 2000. *Strategic Safety Stocks in Supply Chains*, *Lecture Notes in Economics and Mathematical Systems*, vol. 490. Springer, Berlin, Heidelberg. doi:10.1007/978-3-642-58296-7.

Ramya, R. 2024. Quantum computing in inventory management and dynamic allocation. Ahdi Hassan, Pronaya Bhattacharya, Pushan Kumar Dutta, eds., *Quantum Computing and Supply Chain Management: A New Era of Optimization*, chap. 7. IGI Global, Hershey, PA, USA, 99–112. URL `https://www.igi-global.com/book/quantum-computing-supply-chain-management/338306`.

Regev, Oded. 2024. An efficient quantum factoring algorithm. *arXiv:2308.06572v3* URL `http://arxiv.org/abs/2308.06572v3`.

Sharma, Monit, Hoong Chuin Lau. 2024. Quantum monte carlo methods for newsvendor problem with multiple unreliable suppliers. *arXiv preprint* **quant-ph**(2409.07183v1). URL `https://arxiv.org/abs/2409.07183v1`.

Sharma, Monit, Hoong Chuin Lau, Rudy Raymond. 2024. Quantum-enhanced simulation-based optimization for newsvendor problems. *arXiv preprint* **quant-ph**(2411.11756). URL `https://arxiv.org/abs/2411.11756`.

Shor, Peter W. 1996. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *arXiv:quant-ph/9508027v2* URL `http://arxiv.org/abs/quant-ph/9508027v2`.

Xie, Ningyi, Xinwei Lee, Dongsheng Cai, Yoshiyuki Saito, Nobuyoshi Asai, Hoong Chuin Lau. 2024. A feasibility-preserved quantum approximate solver for the capacitated vehicle routing problem. *arXiv preprint* **quant-ph**(2308.08785v3). URL `https://arxiv.org/abs/2308.08785`.

Appendix

1 Appendix A - Obtaining the QUBO Model

In this section, we present the calculation process of obtaining the general QUBO model that can be implemented in a quantum circuit.

The original general model of the guaranteed service time safety stock allocation problem with identical service times is

Model
objective

$$min \quad C = \sum_{i=1}^{n} h_i k_i(SL_i)\sqrt{X_i} \tag{1}$$

subject to

$$\max_{w \in W(A,i)} \left\{ \sum_{j \in w} (\lambda_j - T_j) \right\} \geq 0, \qquad \forall i = 1, 2, \ldots, n \tag{2}$$

$$\sum_{j \in w} T_j \geq \sum_{j \in w} \lambda_j - \sum_{i \in \{w \cap E\}} \overline{ST}_i, \qquad \forall w \in W(A, E) \tag{3}$$

$$X_i = \sigma_i^2 T_i, \quad \text{with} \quad \sigma_i^2 = \sum_{e \in E(i)} a_{ie}^2 \sigma_e^2 \tag{4}$$

$$T_i = \max_{j \in v(i)} \{ST_j\} + \lambda_i - ST_i \qquad \forall i = 1, 2, \ldots, n \tag{5}$$

$$T_i, ST_i, X_i \geq 0, \qquad \forall i = 1, 2, \ldots, n \tag{6}$$

Firstly, we apply the following substitution

$$\sqrt{X_i} = \sigma_i \sqrt{T_i}$$

so the objective function becomes

$$min \quad C = \sum_{i=1}^{n} h_i k_i(SL_i)\sigma_i\sqrt{T_i} \tag{7}$$

Knowing that

$$z_i = \sqrt{T_i}$$

or

$$z_i^2 = T_i$$

the resulting model obtained is now

Model
objective

$$min \quad C = \sum_{i=1}^{n} h_i k_i(SL_i)\sigma_i z_i \tag{8}$$

subject to

$$\max_{w\in W(A,i)}\left\{\sum_{j\in w}(\lambda_j - z_j^2)\right\} \geq 0, \qquad \forall i = 1,2,\dots,n \tag{9}$$

$$\sum_{j\in w} z_j^2 \geq \sum_{j\in w}\lambda_j - \sum_{i\in\{w\cap E\}}\overline{ST}_i, \qquad \forall w \in W(A,E) \tag{10}$$

$$z_i = \max_{j\in v(i)}\{ST_j\} + \lambda_i - ST_i \qquad \forall i = 1,2,\dots,n \tag{11}$$

$$z_i, ST_i \geq 0, \qquad \forall i = 1,2,\dots,n \tag{12}$$

Now, we need to linearize the constraints (9) and (11).
To do that, we first highlight that the constraint (9) for a specific i is met if just one one of the $\max_{w\in W(A,i)}\left\{\sum_{j\in w}(\lambda_j - z_j^2)\right\}$ values satisfy the constraint. So we rewrite it as

$$u_i \leq \sum_{j\in w_k}(\lambda_j - z_j^2) + M(1 - y_{k,i}) \qquad \forall i = 1,2,\dots,n, \forall k \in |W(A,i)| \tag{13}$$

$$\sum_{k=1}^{|W(A,i)|} y_{k,i} = 1 \qquad \forall i = 1,2,\dots,n \tag{14}$$

$$u_i \geq 0 \qquad \forall i = 1,2,\dots,n \tag{15}$$

$$y_{k,i} \in \{0,1\} \tag{16}$$

where

M is a Big-M notation parameter

u_i are slack variables to help with the linearization

$y_{k,i}$ are auxiliary binary variables

Regarding the constraint (11), the linearization is quite similar

$$z_i^2 = H_i + \lambda_i - ST_i \qquad \forall i = 1,2,\dots,n \tag{17}$$

$$H_i \geq ST_j \qquad \forall i = 1,2,\dots,n, \forall j \in v(i) \tag{18}$$

$$H_i \leq ST_j + N(1 - x_{i,j}) \qquad \forall i = 1,2,\dots,n, \forall j \in v(i) \tag{19}$$

$$\sum_{j\in v(i)} x_{i,j} = 1 \qquad \forall i = 1,2,\dots,n \tag{20}$$

$$x_{k,i} \in \{0,1\} \tag{21}$$

where

M is a Big-M notation parameter

H_i are slack variables to help with the linearization

$y_{k,i}$ are auxiliary binary variables

Now, we can write the Lagrangian formulation of the obtained model

$$
\begin{aligned}
min\ C = \sum_{i=1}^{n} h_i \Phi_{0,1}^{-1}\left(SL_i\right)\sigma_i z_i + \\
+ \alpha \sum_{i=1}^{n} \left(1 - \sum_{k=1}^{|W(A,i)|} y_{k,i}\right)^2 + \\
+ \beta \sum_{i=1}^{n} \sum_{k=1}^{|W(A,i)|} \left(u_i - \sum_{j \in w_k} (\lambda_j - z_j^2) - M(1 - y_{k,i}) + P_{k,i}\right)^2 + \\
+ \gamma \left(\sum_{i=1}^{n} \lambda_i - \sum_{i \in \{w \cap E\}} ST_i - \sum_{i=1}^{n} z_i^2 + Q\right)^2 + \\
+ \rho \sum_{i=1}^{n} \left(z_i^2 - H_i - \lambda_i + ST_i\right)^2 + \\
+ \eta \sum_{i=1}^{n} \sum_{j \in v(i)} \left(ST_j - H_i + R_{i,j}\right)^2 + \\
+ \epsilon \sum_{i=1}^{n} \sum_{j \in v(i)} \left(H_i - ST_j - N(1 - x_{i,j}) + U_{i,j}\right)^2 + \\
+ \delta \sum_{i=1}^{n} \left(1 - \sum_{j \in v(i)} x_{i,j}\right)^2
\end{aligned}
\tag{22}
$$

where

$z_i = \sqrt{T_i}$

α, β, γ, ρ, η, ϵ and δ are the introduced Lagrangian penalty terms

$y_{k,i}$ and $x_{i,j}$ are auxiliary binary variables

$V_{k,i,b}$, Q_b, $R_{i,j,b}$ and $U_{i,j,b}$ are slack variables used to transform inequality constraints of the original model to equality constraints.

H_i and u_i are variables used to linearized the original *max* constraints

M and N are "Big-M" notation parameters.

The only issue now is that the variables z_i, $V_{k,i,b}$, Q_b, $R_{i,j,b}$, $U_{i,j,b}$ H_i and u_i are continuous/integer variables, but the QUBO formulation requires only binary variables. To overcome this issue, we need to bind the possible values of these variables to integers. Given that, we can transform the integer variables into binary variables by performing integer-to-binary encoding. With this technique, integer variable values are split into their single bits.
By doing that the final QUBO model is

$$
\begin{aligned}
\min C = \sum_{i=1}^{n} h_i\, \Phi_{0,1}^{-1}(SL_i)\ \sigma_i \left(\sum_b 2^b z_{i,b} \right) + \\
+ \alpha \sum_{i=1}^{n} \left(1 - \sum_{k=1}^{|W(A,i)|} y_{k,i} \right)^2 + \\
+ \beta \sum_{i=1}^{n} \sum_{k=1}^{|W(A,i)|} \Bigg(\sum_b 2^b u_{i,b} - \sum_{j \in w_k} \Big(\lambda_j - \Big(\sum_b 2^b z_{j,b} \Big)^2 \Big) \\
- \Big(\sum_b 2^b M_b \Big) \big(1 - y_{k,i}\big) + \sum_b 2^b V_{k,i,b} \Bigg)^2 + \\
+ \gamma \left(\sum_{i=1}^{n} \lambda_i - \sum_{i \in \{w \cap E\}} \sum_b 2^b ST_{i,b} - \sum_{i=1}^{n} \left(\sum_b 2^b z_{i,b} \right)^2 + \sum_b 2^b Q_b \right)^2 + \\
+ \rho \sum_{i=1}^{n} \left(\left(\sum_b 2^b z_{i,b} \right)^2 - \sum_b 2^b H_{i,b} - \lambda_i + \sum_b 2^b ST_{i,b} \right)^2 + \\
+ \eta \sum_{i=1}^{n} \sum_{j \in v(i)} \left(\sum_b 2^b ST_{j,b} - \sum_b 2^b H_{i,b} + \sum_b 2^b R_{i,j,b} \right)^2 + \\
+ \epsilon \sum_{i=1}^{n} \sum_{j \in v(i)} \left(\sum_b 2^b H_{i,b} - \sum_b 2^b ST_{j,b} - \Big(\sum_b 2^b N_b \Big) \big(1 - x_{i,j}\big) + \sum_b 2^b U_{i,j,b} \right)^2 + \\
+ \delta \sum_{i=1}^{n} \left(1 - \sum_{j \in v(i)} x_{i,j} \right)^2 .
\end{aligned}
\tag{23}
$$

where

b identify the b^{th} bit of an integer variable.

$z_i = \sqrt{T_i}$

α, β, γ, ρ, η, ϵ and δ are the introduced penalty terms

$y_{k,i}$ and $x_{i,j}$ are auxiliary binary variables

$V_{k,i,b}$, Q_b, $R_{i,j,b}$ and $U_{i,j,b}$ are slack variables used to transform inequality constraints of the original model to equality constraints.

H_i, b and u_i, b are variables used to linearized the original *max* constraints

M and N are "Big-M" notation parameters.

Ehrenwörtliche Erklärung

Ich erkläre hiermit ehrenwörtlich, dass ich die vorliegende Arbeit selbständig angefertigt habe. Die aus fremden Quellen direkt und indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Ich weiß, dass die Arbeit in digitalisierter Form daraufhin überprüft werden kann, ob unerlaubte Hilfsmittel verwendet wurden und ob es sich - insgesamt oder in Teilen - um ein Plagiat handelt. Zum Vergleich meiner Arbeit mit existierenden Quellen darf sie in eine Datenbank eingestellt werden und nach der Überprüfung zum Vergleich mit künftig eingehenden Arbeiten dort verbleiben. Weitere Vervielfältigungs- und Verwertungsrechte werden dadurch nicht eingeräumt. Die Arbeit wurde weder einer anderen Prüfungsbehörde vorgelegt noch veröffentlicht.

Ort, Datum

Unterschrift