Quantum Computing Methods for Supply Chain Management

Hansheng Jiang∗ Zuo-Jun Max Shen†‡§ Junyu Liu‡§
∗Dept. of Industrial Engineering and Operations Research
University of California, Berkeley
Berkeley, CA 94706, USA
hansheng_jiang@berkeley.edu

†Dept. of Industrial Engineering and Operations Research
University of California, Berkeley
Berkeley, CA 94706, USA
†Faculty of Engineering and Faculty of Business and Economics
University of Hong Kong
Hong Kong, China
maxshen@berkeley.edu

‡§qBraid Co.
Chicago, IL 60615, USA
junyuli@uchicago.edu

Abstract—Quantum computing is expected to have transformative influences on many domains, but its practical deployments on industry problems are underexplored. We focus on applying quantum computing to operations management problems in industry, and in particular, supply chain management. Many problems in supply chain management involve large state and action spaces and pose computational challenges on classic computers. We develop a quantized policy iteration algorithm to solve an inventory control problem and demonstrate its effectiveness. We also discuss in-depth the hardware requirements and potential challenges on implementing this quantum algorithm in the near term. Our simulations and experiments are powered by IBM Qiskit and the qBraid system.

Index Terms—quantum computing, supply chain management, policy iteration, quantum linear system solver

I. INTRODUCTION

Recent surveys indicate promising prospects of quantum computing in many fields, for example, in finance [1], [2] and quantum chemistry simulation [3]. In this work, we explore what advantages quantum computing may provide for addressing important problems in the field of operations management.

Supply chain management is a central problem in the field of operations management. Supply chain management is a discipline studying the flow of goods or services through all the different stages of the process. From the early days when inventory decisions were manually documented with pen and paper, supply chains have undergone a major transformation to be more automated and efficient, thanks to the advances of new technologies such as digitalization, software development, and more computing power. Therefore, it is natural to wonder what the technology of quantum computing may offer to supply chain management. In particular, it is well known that the large decision spaces of inventory control problems brought major computational challenges to supply chain management, and some recent studies have explored using deep learning methods for supply chain management [4].

In this work, we conduct a modest inquiry of the prospect of quantum computing on supply chain management by focusing on a classic inventory control problem. We describe special features of this inventory control problem that make quantum computing a suitable tool. We also discuss the practical limitations of certain quantum approaches in terms of hardware requirements and error correction. We run numerical experiments on IBM Qiskit [5] and the qBraid system [6] to establish the validity.

II. INVENTORY CONTROL PROBLEM

We consider a classic periodic-review inventory control problem in supply chain management [7]. At each time period, the manager of a warehouse needs to decide on the number of product units to order. We assume that there is no lead time and the ordering cost is proportional to the number of units ordered. At each time period, the demand is stochastic following some probability distribution known to the manager. When there is no sufficient inventory, excess demand is lost. The manager needs to make ordering decisions to balance the ordering cost and lost sales cost in the long run.

This inventory control problem can be naturally modeled by the Markov Decision Process. Let $s_t$ represent the inventory level at the beginning of the $t$-th time period, and $a_t$ represent the number of units ordered by the inventory manager in the beginning of the period $t$. Correspondingly, the state space of $s_t$ and action space of $a_t$ are denoted by $S = \{1, \ldots, |S|\}$ and $A = \{1, \ldots, |A|\}$, respectively. Let $D_t$ represent the stochastic demand of this time period. We assume that the probability distribution of demand is stationary over time and can be specified by values $p_d = P(D_t = d)$ for $d = 1, 2, \ldots, D$ for some integer $D > 0$. The inventory level in the beginning of $t + 1$, denoted by $s_{t+1}$, is captured by the following system transition function,

$$s_{t+1} = [s_t + a_t - D_t]^+,$$

(1)
where \( x^+ := \max\{x, 0\} \). The sequence of events is illustrated in Fig. 1.

![Sequence of events in the inventory control problem.](image)

**Algorithm 1** Policy Iteration Algorithm for Inventory Control

**Input:** demand distribution, cost parameters, \( K \)

**Output:** optimal inventory reorder policy

1. **Initialization:** initial policy \( \pi_0 \)
2. Calculate \( Q^{\pi_k} \in \mathbb{R}^{|S|} \) by solving a linear system
   \[ Q^{\pi_t} = (I - \gamma P^{\pi_{t-1}})^{-1} r \]
3. Obtain new policy \( \pi_{k+1} \) by greedily solving the following maximization with respect to \( v^{\pi_k}_s \): for all \( i \in S \)
   \[ \pi_{k+1}(i) = \arg \max_{j \in A} Q^{\pi_k}(i, j) \]
4. **end for**
5. **return** \( \pi_K \)

We have chosen the termination condition of Algorithm 1 by setting the total number of iterations as some large enough number \( K \) because it is known that policy iteration algorithm can converge in finite iterations [11]. In Step 2 of Algorithm 1, we use quantum linear system solver to provide quantum states to approximate the Q function \( Q^{\pi_k} \). Let \( B_k \) denote \( B_k := I - \gamma P^{\pi_k} \), then the policy iteration step is finding the value of \( B_k^{-1} r \), where the dimension of the matrix \( B_k \) is \( |S||A| \).

In Step 3, the new policy \( \pi_{k+1}(i) \) is obtained by finding the position of the largest element in \( Q^{\pi_k}(i, j) \) for the fixed \( i \).

A key observation is that the large matrices \( B_k \) and \( P^{\pi_k} \) have some sparsity properties for any deterministic policy \( \pi \) under a very practical sparsity assumption on the inventory demand. Based on the definition in (4), for every given \((i, j)\), \( P_{(i', j), (i, j')}^{\pi_k} \neq 0 \) only when \( j' = \pi(i') \) and \( P(i'|i, j) > 0 \). Assuming that the demand only takes \( r \leq |S| \) different values, then the number of non-zeros in \( P^{\pi_k}_{i,j} \) is bounded by \( r \times |S||A| \), and consequently the number of non-zeros in \( B_k \) is bounded by \( (r + 1) \times |S||A| \), which is much smaller than the total number of elements \( |S|^2|A|^2 \).

A. Preliminaries on Policy Iteration

A classic algorithm for solving the Markov Decision Process is the policy iteration algorithm [10], which generates a sequence of policies that gradually converge to the optimal solution over iterations. The policy iteration algorithm enjoys nice convergence properties and actually terminates within finite iterations. Specifically, it is proved in [11] that the number of iterations of policy iteration can be bounded in polynomial time \( O((|S||A| - |S|)/(1-\gamma)) \), where \( |S| \) and \( |A| \) stand for the size of state space and action space, respectively. However, each iteration of the policy iteration algorithm is time-consuming due to the sizes of the state space and the action space. The per iteration complexity of the policy iteration algorithm has a time complexity of \( |S|^3 + |S|^2|A| \). It leaves space to leverage quantum computing to implement the policy iteration algorithm.
The task now is to find the minimal eigenvalue of the Hermitian operator $O$, which might be obtained by traditional gradient descent algorithms,

$$\theta_i(t+1) - \theta_i(t) = -\eta \frac{\partial L}{\partial \theta_i},$$

with the learning rate $\eta$ and the number of iterations $t$. In supervised learning, we could encode supervised data in $|\Psi_0\rangle$. Thus, supervised learning could be performed with quantum machines similarly.

One could give a variational version of the HHL algorithm by simply preparing variational states $|x(\theta)\rangle = U(\theta)|\Psi_0\rangle$ such that one could minimize the difference between $A|x(\theta)\rangle$ and $b$, in order to solve the matrix inversion of $A$. We adapt the variational ansatz in [15], where people find a comparable performance against HHL with shorter circuit depth. We will primarily study the implementation of it in the near-term simulation and benchmarks.

IV. QRAM HARDWARE REQUIREMENTS

When applying HHL-like algorithm, one of the primary challenges is to inform quantum computers the matrix elements of $A$ we want to invert. This problem could be solved, in principle, using so-called Quantum Random Access Memory [19] (QRAM). With the data size $N$, QRAM could use $O(N)$ qubits, and $O(\log N)$ time to implement the following unitary operator using a parallel way,

$$\sum_i \alpha_i |i\rangle |0\rangle \rightarrow \sum_i \alpha_i |i\rangle |x_i\rangle$$

where $\alpha_i$’s are arbitrary coefficients and $x_i$’s are the data. In practice, large-scale, fault-tolerant QRAMs are challenging to build [20]–[22]. However, it will be visionary to estimate how hard it is in practice to implement QRAM and HHL for given physical devices and algorithm requirements.

In Figure 2 and 3, we study how many physical parameters are needed for given requirements from current Supply Chain Management. Here, we make use of the hardware models of QRAM [20], [21]. Assuming that the size of the matrix is $10^{13}$, with the precision $10^{-3}$, which could typically happen in the current Supply Chain Management technology, we bound the required decoherence rate $\kappa + \gamma$ in the hybrid Circuit quantum electrodynamics system for realizing QRAM. A general conclusion is that it might be challenging to realize those requirements with the current physical devices, it will be helpful to study possibilities of realizing those experiments in the long term.

Finally, we comment that the classic-quantum interfaces are essential for HHL-based algorithms. QRAM architectures solve the uploading problem, and downloading the quantum data towards classic devices could be done using classic shadow [23], which is exponentially efficient for sparse quantum state tomography.

In Figure 2, we calculate the error rate from the precision of the problem $1 - F$ and the size of the data $N$, with the assumption that $T = \log N$. The region bounded by the red
box indicates the current constraints used in the Supply Chain Optimization community. We use the infidelity formula proven from [21], $1 - F \sim \frac{1}{2} \varepsilon T \log N \sim \frac{1}{4} \varepsilon^2 \log^2 N$ for QRAM architectures. The color density is the error rate $\varepsilon$.

In Figure 3, we assume $g_d = 1 \text{ kHz} \times 2 \pi$, $\nu = 10 \text{ MHz} \times 2 \pi$, and $\nu_d = 4.5$ which is the average of the CZ and SWAP gates inside the QRAM circuit as an estimate, in the setup of [20] with the formula $\varepsilon \approx (\kappa + \gamma) \frac{\gamma_{\text{phon}}}{2 \pi g_d} + \left(\frac{\nu_d}{\nu}\right)^2$. Here, $\kappa$ and $\gamma$ are the phonon and transmon decoherence rates, $\nu$ is the free spectral range, and $g_d$ is the direct coupling. We give an emphasis on the size of the modern Supply Chain Management problem in the red line. The color density is the error rate $\varepsilon$.

V. SIMULATION EXPERIMENTS

In this section, we give precise simulation details about how to solve our Markov Decision Process using quantum computing.

Precisely speaking, we implement the policy iteration step (Step 2) of Algorithm 1 in quantum computers. Step 2 is essentially solving a large linear system in (5), and we use the variational quantum algorithms introduced in Section III-C. Under the simplified notation, $B := (I - \gamma P^\pi)$ and $q := Q^\pi$, we need to solve the following linear system

$$Bq = r,$$

where the dimension of $B$ is $|S||A|$.

A. LCU Coefficients

We use the oracle model, so-called the LCU (Linear Combination of Unitaries) decomposition to upload the data of the matrix $B$ we need to invert in each iteration of the Markov Decision Process. The LCU decomposition is defined as

$$B = \sum_{i=1}^{L} a_i P_i,$$

where $a_i$’s are real coefficients (since $B$ is Hermitian) and $P_i$’s are unitary operations. One could choose $P_i$ as elements of the Pauli group ($4^N$ elements in total with $N$ qubits if we do not count for redundancies from signs), one could compute the coefficients $a_i$ according to

$$a_i = \frac{1}{2^N} \text{Tr}(B P_i).$$

Figure 4 shows the distribution of the LCU coefficients of a single matrix instance used in the Markov Decision Process.
implemented with the help of fault-tolerant quantum computing.

### Table I

| $L = 1^2$ | $L = 2^2$ | $L = 3^2$ | $L = 4^2$ |
|-----------|-----------|-----------|-----------|
| $N = 1$   | 75        | ×         | ×         |
| $N = 2$   | 273       | 274       | 575       | 2215      |
| $N = 3$   | 613       | 1500      | 3110      | 5086      |
| $N = 4$   | 1161      | 3090      | 6466      | 10071     |
| $N = 5$   | 2672      | 8388      | 14301     | 19329     |
| $N = 6$   | 2680      | 5898      | 13290     | 28143     |

In Table I, we compute the number of fundamental gates used in IBM Qiskit for the HHL oracle for the matrix inversion task in the Markov Decision Process. We use $\times$ to denote the disallowed situation where the dimension of the Hilbert space cannot hold so much independent $L$. Moreover, in order to scale, we choose the leading submatrix with the corresponding size by given number of qubits $N$ and the number of the LCU terms $L$.

### C. Variational Circuits

We use the IBM Qiskit system and the qBraid system to simulate the variational quantum algorithms for linear system solving, with the Markov Decision Process. With the qBraid environment, we compare the simulations between noiseless and noisy environments provided by IBM Qiskit with the real hardware noise models, with first five truncated LCU decomposition coefficients and 6 qubits, in Figure 5. We find that the variational simulation could converge in a decent amount of time even in the noisy environment.

![Fig. 5. Solving the matrix inversion problem by variational quantum linear solver.](image)

In Figure 5, we use the variational ansatz provided in [15] with 2 layers and 6 qubits. The noise calculation is from the real quantum device model in IBM Qiskit.

### VI. Conclusion

Despite the promising future of quantum computing, more efforts are needed to make it practical in solving real-world problems. This work explores the usage of quantum computing in the field of supply chain management by focusing on a canonical inventory control problem. We discuss in-depth a classic inventory control problem and propose to solve it with a quantized policy iteration algorithm. Our experiments on IBM Qiskit and the qBraid system demonstrate the practicality of variational algorithms for solving small-sized inventory control problems. We believe this is a well open area that will be interesting for both academia and industry to explore further.

**Acknowledgment**

We thank three anonymous reviewers for constructive and helpful feedbacks.

### References

[1] R. Orus, S. Mugel, and E. Lizaso, “Quantum computing for finance: Overview and prospects,” Reviews in Physics, vol. 4, p. 100028, 2019.

[2] D. Herman, C. Googin, X. Liu, A. Galda, I. Safro, Y. Sun, M. Pistoia, and Y. Alexeev, “A survey of quantum computing for finance,” arXiv preprint arXiv:2201.02773, 2022.

[3] B. Bauer, S. Bravyi, M. Motta, and G. K.-L. Chan, “Quantum algorithms for quantum chemistry and quantum materials science,” Chemical Reviews, vol. 120, no. 22, pp. 12685–12717, 2020.

[4] J. Gisbrechts, R. N. Boute, J. A. Van Mieghem, and D. Zhang, “Can deep reinforcement learning improve inventory management? Performance on dual sourcing, lost sales and multi-echelon problems,” Manufacturing & Service Operations Management, 2021.

[5] https://qiskit.org.

[6] https://qbird.com.

[7] L. V. Snyder and Z.-J. M. Shen, Fundamentals of Supply Chain Theory. Wiley Online Library, 2011.

[8] M. L. Puterman, Markov decision processes: discrete stochastic dynamic programming. John Wiley & Sons, 1994.

[9] A. Agarwal, N. Jiang, S. M. Kakade, and W. Sun, “Reinforcement learning: Theory and algorithms,” CS Dept., UW Seattle, Seattle, WA, USA, Tech. Rep, pp. 10–4, 2019.

[10] R. A. Howard, Dynamic programming and markov processes. John Wiley, 1960.

[11] Y. Ye, “The simplex and policy-iteration methods are strongly polynomial for the markov decision problem with a fixed discount rate,” Mathematics of Operations Research, vol. 36, no. 4, pp. 593–603, 2011.

[12] A. W. Harrow, A. Hassidim, and S. Lloyd, “Quantum algorithm for linear systems of equations,” Physical review letters, vol. 105, no. 15, p. 150502, 2009.

[13] A. Scherer, B. Valiron, S.-C. Mau, S. Alexander, E. Van den Berg, and T. E. Chapuran, “Concrete resource analysis of the quantum linear-system algorithm used to compute the electromagnetic scattering cross section of a 2d target,” Quantum Information Processing, vol. 16, no. 3, pp. 1–65, 2017.

[14] J. Preskill, “Quantum Computing in the NISQ era and beyond.” Quantum, vol. 2, p. 79, Aug. 2018.

[15] C. Bravo-Prieto, R. LaRose, M. Cerezo, Y. Subasi, L. Cincio, and P. J. Coles, “Variational quantum linear solver,” arXiv preprint arXiv:1909.05820, 2019.

[16] H.-Y. Huang, K. Bharti, and P. Rebentrost, “Near-term quantum algorithms for linear systems of equations with regression loss functions,” arXiv preprint arXiv:2210.02773, 2022.
[20] C. T. Hann, *Practicality of Quantum Random Access Memory*. PhD thesis, Yale University, 2021.

[21] C. T. Hann, G. Lee, S. Girvin, and L. Jiang, “Resilience of quantum random access memory to generic noise,” *PRX Quantum*, vol. 2, no. 2, p. 020311, 2021.

[22] K. C. Chen, W. Dai, C. Errando-Herranz, S. Lloyd, and D. Englund, “Scalable and high-fidelity quantum random access memory in spin-photon networks,” *PRX Quantum*, vol. 2, no. 3, p. 030319, 2021.

[23] H.-Y. Huang, R. Kueng, and J. Preskill, “Predicting many properties of a quantum system from very few measurements,” *Nature Physics*, vol. 16, no. 10, pp. 1050–1057, 2020.