Hybrid Quantum-Classical Algorithms for Loan-Collection Optimization with Loan-Loss Provisions

Jirawat Tangpanitanon,[1,2,*] Jirawat Saiphet,[3,†] Pantita Palittapongarnpim,[4]
Poompong Chaiwongkhot,[3] Pinn Prugsanapan,[5] Nuntanut Raksasri,[5] Wipada Wannasiwaporn,[5]
Yarnvith Raksri,[5] Pairash Thajchayapong,[5] and Thiparat Chotibut[4,‡]

[1] *Quantum Technology Foundation (Thailand), Bangkok, Thailand*
[2] *Thailand Center of Excellence in Physics, Ministry of Higher Education, Science, Research and Innovation, Bangkok, Thailand*
[3] *Optical and Quantum Physics Laboratory, Department of Physics, Faculty of Science, Mahidol University, Bangkok, Thailand*
[4] *Chula Intelligent and Complex Systems Lab, Department of Physics, Faculty of Science, Chulalongkorn University, Bangkok, Thailand*
[5] *Kasikorn Business-Technology Group, Bangkok, Thailand*

(Received 7 February 2023; accepted 4 May 2023; published 1 June 2023)

Banks are required to set aside funds in their income statement, known as a loan-loss provision (LLP), to account for potential loan defaults and expenses. By treating the LLP as a global constraint, we propose a hybrid quantum-classical algorithm to solve a specific case of quadratic constrained binary optimization (QCBO) models for loan-collection optimization. The objective is to find a set of optimal loan-collection actions that maximizes the expected net profit presented to the bank as well as the financial welfare in the financial network of loanees, while keeping the LLP at its minimum. Our algorithm consists of three parts: a classical divide-and-conquer algorithm to enable a large-scale optimization, a quantum alternating operator ansatz (QAOA) algorithm to maximize the objective function, and a classical sampling algorithm to handle the LLP. We apply the algorithm to a real-world data set with 600 loanees and five possible collection actions. The QAOA is performed using up to 35 qubits on a classical computer. We show that incorporating the QAOA can enhance the expected net profit by approximately 70% in comparison to scenarios where the QAOA is absent from the hybrid algorithm. Although this improvement does not constitute definitive evidence of quantum advantage, our work illustrates the use of near-term quantum devices to tackle real-world optimization problems.

DOI: 10.1103/PhysRevApplied.19.064001

I. INTRODUCTION

Financial institutions often rely on the ability to solve computationally intensive problems. Quantum computing holds the promise of highly efficient algorithms that can provide speedup over some best-known classical algorithms [1–3]. Applications of quantum computing on finance have been recently explored both theoretically and experimentally on small quantum devices. Examples include portfolio optimization [4–7], option pricing [8–10], risk analysis [11], and credit valuation adjustment [12]. More comprehensive overviews of the emerging field of quantum computing in finance can be found in Refs. [1,3,13].

In this work, we introduce an alternative use case of quantum computing in finance, namely loan-collection optimization. The goal is to find a set of optimal actions to be taken on the loanees in order to maximize the expected net profit presented to the lender. Examples of these actions include doing nothing, creating a promise-to-pay agreement, debt restructuring and offering various forms of discounted payoffs (DPOs) [14]. DPOs typically occur in distressed loan scenarios, where loanees are experiencing financial or operational distress, default, or are under bankruptcy. Usually, DPOs are a last resort for lenders because they often involve taking a loss as the loan is repaid for less than the outstanding balance.

Loan-collection actions are normally chosen based on the history of an individual loanee and the experience of the collector. However, the network-induced "domino" effects of such actions, where financial distress can potentially cascade through interconnected financial network of

*jirawat@qtft.org
†jirawat.sai@outlook.com
‡Corresponding author. thiparat.c@chula.ac.th

loanees, are usually ignored due to the complexity of the underlying financial system [15]. For example, offering a DPO to one loanee usually signifies the cash-flow problem of such loanee, which could cause disruptions in the supply chain [16]. Such scenarios would negatively impact cash flow of related parties. Some of which may also be the customer of the lender. Therefore, it is useful to account for financial association among the loanees when determining the optimal collection actions. The lender should aim to maximize the expected net profit while minimizing the potential cascade of financial distress on the financial network of the loanees.

Another aspect of loan collection is loan-loss provision (LLP) [17]. LLP is regulated by the government to ensure the financial health of the bank. The idea is to set aside funds as an expense in the financial statement to account for potential loan defaults and expenses that occur as a result of lending. The amount of LLPs depends on various factors such as types of loanees (individuals, small businesses, large corporations), types of loans, late payments, collection expenses, as well as the collection actions that will be taken. LLPs typically constitute a significant portion of the financial statement. For example, in the first half of 2020, seventy banks worldwide report under expected credit loss accounting provisions totaling in \$161 billion [18]. Therefore, it is crucial to choose optimal collection actions that keep LLPs at a minimum to promote financial liquidity.

Here, we model the above problem including the financial network and LLPs as quadratic constrained binary optimization (QCBO) problems with both global and local constraints. We solve this class of problems by devising a hybrid quantum-classical algorithm, which consists of three parts: a classical divide-and-conquer algorithm to enable a large-scale optimization, a quantum alternating operator ansatz (QAOA) algorithm to maximize the objective function, and a classical sampling algorithm to handle the LLP. By benchmarking with a real-world data set, we show that the presence of the QAOA can improve the expected net profit by approximately 70%, compared to when the QAOA is absent from the hybrid algorithm. This increase is not a proof of a quantum advantage in general, but rather an evidence of a promising use case for QAOA that calls for further investigation. Our work can be implemented on near-term quantum devices. For example, the XY terms in the mixing Hamiltonian naturally arise in superconducting-qubit systems via capacitive coupling [19]. In addition, the association matrix, which appears in the coupling terms in the encoding Hamiltonian is usually sparse due to the nature of the real-world problem.

II. PROBLEM FORMULATION

Loan collection is a management problem that involves identifying which collection actions to be taken to which loanees at what time. Due to the complexity of the process, it is a common practice to follow a rule-based guideline for collection activities. However, the actions and time can also be personalized, as is the recent approach to loan collection [20,21], in order to respond to the loanee's unique financial history. Moreover, as discussed earlier, one collection action taken to a loanee can affect the financial welfare of other loanees in the financial network, which, in turn, affects the expected net profit of the lender.

To capture both the personalized loan-collection approach and the effect of loanee's financial network, we propose a heuristic QCBO model that captures the impact of loan-collection actions at a given time for a given loan product. The objective (yield) function is defined as

$$Y = (1-\epsilon)\sum_{i=1}^{N}\sum_{j=1}^{M} h_{i,j}x_{i,j} + \epsilon\sum_{\langle i,i'\rangle}^{N} A_{i,i'}(1-x_{i,1})(1-x_{i',1}), \tag{1}$$

where N is the number of loanees and M is the number of possible actions. Here, $\{x_{i,j}\}$ are binary decision variables: $x_{i,j} = 1$ if action j is taken to loanee i and $x_{i,j} = 0$ otherwise. By convention, we set $j = 1$ to denote the DPO action. The coefficient $h_{i,j} \in \mathbb{R}$ is an expected net profit that the bank would get if action j is taken to loanee i. (The coefficient can also be seen as a way of ranking the bank's preferred action towards a loanee and can be personalized to the loanee's loan payment history. One approach for the personalized calculation of $h_{i,j}$ is to model the loan-collection problem as a Markov decision process (MDP) [20].) $(A_{i,i'}) \in \mathbb{R}^{N\times N}$ is an association matrix, defined as the averaged transaction between loanees i and i'.

According to the form of Y, association or "cash flow" between loanees i and i' vanishes if one of them is offered a DPO, i.e., $x_{i,1} = 1$ or $x_{i',1} = 1$. The hyperparameter $\epsilon \in [0, 1)$ tunes the competition between the expected return to the bank [first term in Eq. (1) without $1-\epsilon$] and the financial welfare, i.e., the average financial transaction within the network excluding the bank [second term in Eq. (1) without ϵ]. Increasing ϵ will cause the optimal solution to contain a lesser number of DPO actions being taken, which increases the overall financial welfare of the loanees.

We now introduce local constraints that force only one action to be taken to each loanee as

$$\sum_{j=1}^{M} x_{i,j} = 1, \quad \forall i \in \{1,\ldots,N\}. \tag{2}$$

LLP is treated as a global constraint as

$$\sum_{i=1}^{N}\sum_{j=1}^{M} l_{i,j}x_{i,j} \leq L, \tag{3}$$

where $L \in \mathbb{R}^+$ is the upper bound for the total provision and $l_{i,j} \in \mathbb{R}^+$ is the provision for taking action j to loanee i. Determining the values of $l_{i,j}$ depends on probability of default estimates and expected loss. Exact details are bank specific and depends on government regulation, such as the Third Basel Accord, an international, voluntary regulatory framework for banks developed by the Basel Committee on Banking Supervision [17].

Although L can be considered a fixed value and a hard constraint, in practice when we are predicting the net income of a known collection of loans, we might allow some leniency in terms of L in favor of a solution that yields high net income. In this scenario, we treat Eq. (3) as a soft constraint with a secondary objective of minimizing the total provision during optimization.

III. HYBRID QUANTUM-CLASSICAL ALGORITHM

In this section, we present our algorithms, as depicted in Fig. 1, to find a set of optimal actions $\{x_{i,j}\}$ that maximizes Y subjected to the constraints in Eqs. (2) and (3). The idea is to use a classical divide-and-conquer algorithm [22] to arrange the loanees into small groups based on the association matrix. Loanees from the same group will have a higher association compared to those outside the group. Next, we find the optimal collection actions for each group using the QAOA algorithm [23], without considering LLPs. Actions from different groups are then combined to reconstruct the optimal actions for the original problem. The latter process is referred to as state reconstruction. Lastly, we apply a classical sampling method, which we call greedy provision reduction (GPR) to adjust some actions to minimize LLPs while keeping the negative impacts of those adjustments on Y at a minimum. We lay out the details of each step below. The pseudocode is provided in Appendix B.

Division algorithm. To divide the loanees into groups, we rely on the use of two standard community detection algorithms, namely the Clauset-Newman-Moore greedy modularity maximization [24] to extract the communities and the Louvain method [25] for further readjustment of communities' membership. After the first iteration, each loanee will be assigned to exactly one group. For example, in Fig. 1, we suppose that the Clauset-Newman-Moore algorithm outputs groups S_1 and S_2, among others. Group S_1 contains loanees $i = 1, 2, 3, 4, 5$. Group S_2 contains loanee $i = 8, 10, 11$. Notice that the two groups have no common nodes. (We note that, in general, the combination of both community detection algorithms will iterate until modularity maximization is convergence, which can result in multiple group partition, as opposed to the example schematic in Fig. 1.) As each group will first be optimized separately, the sparse connection between S_1 and S_2 through loanees $i = 7, 9$ will be ignored. To mitigate this algorithmic artefact in this divide-and-conquer approach, we introduce "edge nodes," which are outsider loanees that have an association with S_1 and S_2. As shown in Fig. 1, the groups S_1, S_2 have edge nodes $i = 6, 7, 9$ and $i = 7, 9$, respectively. By adding the edge nodes into each group, the alternative groups now have loanees $i = 7, 9$ in common. The latter will play a crucial role during state reconstruction. The Louvian method and the edge node introduction are then applied recursively to S_1 and S_2 until the size of each group is no greater than some fixed threshold $\nu \in \mathbb{Z}^+$. The latter ensures that the size of each group will be small enough to be run on near-term quantum devices.

QAOA algorithm. To apply the QAOA, we represent a set of actions applied to the loanees as a basis state of a quantum system consisting of up to $\nu \times M$ qubits. Action j is taken to loanee i ($x_{i,j} = 1$) if qubit (i, j) is in the excited state $|1\rangle$; otherwise, the qubit is in the ground state $|0\rangle$. The QAOA is executed by running the following evolution:

$$|\psi_T(\underline{\theta})\rangle = \prod_{t=1}^{T} e^{-i\hat{H}_A\gamma_t} e^{-i\hat{H}_B\beta_t} |\psi_0\rangle, \tag{4}$$

where T is the number of driving cycles, $\gamma_t, \beta_t \in \mathbb{R}$ are variational parameters and $\underline{\theta} = \{\gamma_1, \beta_1, \ldots, \gamma_T, \beta_T\}$. The Hamiltonian $\hat{H}_A$ involves interactions among qubits with $j = 1$ as labeled by solid gray lines in Fig. 2. It is defined as

$$\hat{H}_A = -(1-\epsilon) \sum_{i=1}^{N'} \sum_{j=1}^{M} h_{i,j} \hat{n}_{i,j} - \epsilon \sum_{\langle i,i' \rangle}^{N'} A_{i,i'} (1 - \hat{n}_{i,1})(1 - \hat{n}_{i',1}). \tag{5}$$

Here, $N' \leq \nu$ is the number of loanees in the group and $\hat{n}_{i,j}$ is the number operator acting on qubit (i, j), which has an eigenvalue 1 if such a qubit is in the excited state $|1\rangle$ and 0 otherwise (one can think of this operator as acting on a Fock's space). The mixing Hamiltonian $\hat{H}_B$ involves ring-type interactions among qubits with the same i as labeled by red dotted lines in Fig. 2. It is defined as

$$\hat{H}_B = -\frac{J}{2} \sum_{i=1}^{N'} \left[\sum_{j=1}^{M} \left(\hat{X}_{i,j}\hat{X}_{i,j+1} + \hat{Y}_{i,j+1}\hat{Y}_{i,j} \right) \right], \tag{6}$$

where $\hat{X}_{i,j}$ and $\hat{Y}_{i,j}$ are Pauli's operators acting on qubit (i, j) and $J = 1$ is the coupling strength. The periodic boundary condition is applied.

The initial state $|\psi_0\rangle$ is prepared as a product state such that action $j = 1$ is taken to every loanee, i.e., $x_{i,j} = \delta_{j,1}$ where $\delta_{j,1}$ is the Kronecker δ. Note that since both $\hat{H}_A$ and $\hat{H}_B$ preserve the number of qubit excitations per loanee,

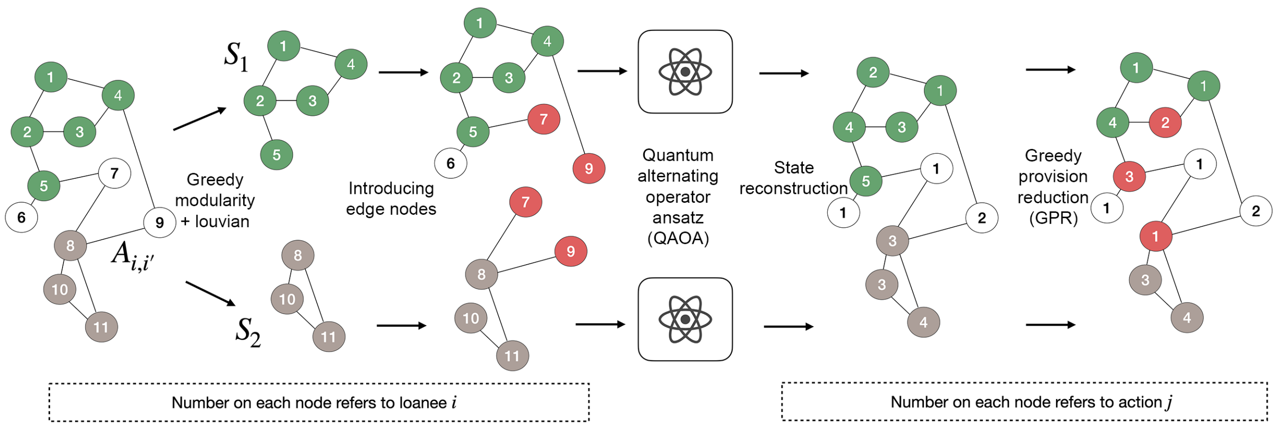

FIG. 1. Schematic of the hybrid quantum-classical algorithm. Firstly, the financial transaction network as indicated by $A_{i,i'}$ is divided into small groups using the greedy modularity algorithm and the Louvian method. The edge nodes, labeled in red, are introduced to each group to accommodate common loanees among the two groups. The optimal set of collection actions for each group is obtained via the QAOA. Solutions from different groups are then combined to reconstruct the optimal actions for the original problem. The latter is fed to the GPR algorithm to minimize LLP. The classical optimization scheme that we use to compare with this hybrid scheme follows the same workflow with the QAOA component removed. See the main text for more details.

it follows that all basis states involved in $|\psi_T(\underline{\theta})\rangle$ always satisfies the constraints in Eq. (2).

Following the standard QAOA procedure, we then readout the observable $\langle \hat{H}_A\rangle_{\underline{\theta}}$. The procedure is repeated with another value of $\underline{\theta}$ as suggested by a classical optimizer until $\langle \hat{H}_A\rangle_{\underline{\theta}}$ is minimized with $\underline{\theta} = \underline{\theta}_{\text{opt}}$. The optimal quantum state $|\psi_T(\underline{\theta}_{\text{opt}})\rangle$ now concentrates around bit strings with high objective function values. The probability of sampling a bit string $\underline{x} = \left[x_{1,1}, x_{1,2}, \ldots, x_{N,M}\right]$ from $|\psi(\underline{\theta}_{\text{opt}})\rangle$ is given by $p(\underline{x}) = |\langle \underline{x}|\psi_T(\underline{\theta}_{\text{opt}})\rangle|^2$, where $|\underline{x}\rangle$ is a basis state representing a set of actions $\underline{x}$.

Reconstruction algorithm. To combine optimal actions from two groups, for example, S_1 and S_2, we simply sample bit strings from $p(\underline{x})$ of each group. If the bit strings from S_1 and S_2 have the same values at the common edge nodes, we append the two bit strings. For example, as depicted in Fig. 1, S_1 has optimal actions $j = 2, 4, 3, 1, 5, 1, 1, 2$ for loanees $i = 1, 2, 3, 4, 5, 6, 7, 9$, respectively, and S_2 has optimal actions $j = 1, 3, 2, 3, 4$ for loanees $i = 7, 8, 9, 10, 11$, respectively. Since the common loanees $i = 7, 9$ have the same actions $j = 1, 2$ in both S_1 and S_2. The optimal actions for the combined group $S_1 \cup S_2$ is $j = 2, 4, 3, 1, 5, 1, 1, 3, 2, 3, 4$ for loanees $i = 1, 2, \ldots, 11$, respectively. If the actions to the common loanees from the two groups are incompatible, then the bit strings are resampled from $p(\underline{x})$ until the compatible one is found. This procedure is repeated for every group until the (locally) optimal actions $\underline{x}^*$ for the original problem with N loanees is obtained.

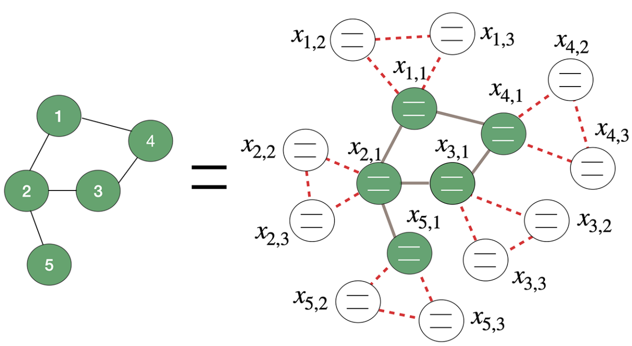

FIG. 2. Qubit representation of loanees. (Left) A group of loanees and their interactions as indicated by $A_{i,i'}$. (Right) The qubit topology to run the QAOA to find optimal collection actions for this set of loanees. Each loanee is represented by M qubits ($M = 3$ in this figure). The interactions among qubits as labeled by solid gray lines and dashed red lines are captured by $\hat{H}_A$ and $\hat{H}_B$, respectively.

Greedy provision reduction (GPR) algorithm. Lastly, we apply a classical sampling algorithm to handle the LLP constraint in Eq. (3) by minimizing the total provision. The idea is to quantify the impact of changing action j of loanee i to j' using a finesse score $f_{i,j\to j'} \in \mathbb{R}$, which is defined by the ratio between the loss in yield Y and the reduction in provision. For a given set of actions $\{i, j\}$, there are $N \times M$ values of $f_{i,j\to j'}$'s, each of which is evaluated as the algorithm explores actions available to loanee i. From this evaluation, we then choose the action that results in the highest $f_{i,j\to j'}$ (greedy) and update $\underline{x}^*$ accordingly.

To define $f_{i,j\to j'}$, we first define a reward $a_{i,j\to j'}$ as a reduction in the total provision and a penalty $b_{i,j\to j'}$ as a reduction in Y, when changing action j of loanee i to action j'. By definition, the finesse score increases when switching actions result in a higher reward and a lower penalty. Specifically, if the total provision does not increase ($a_{i,j\to j'} >= 0$) and Y does not decrease ($b_{i,j\to j'} <= 0$), we choose the action with the highest

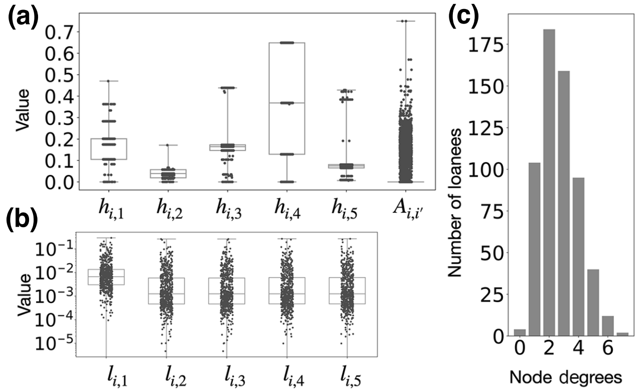

FIG. 3. Statistics of model parameters obtained from Kasikornbank's data set. (a) Box plot of $h_{i,j}$ and $A_{i,i'}$ showing their distributions and quantiles. (b) Box plot of $l_{i,j}$ in a log scale. (c) The histogram of the node degrees in $A_{i,i'}$.

reward, i.e., $f_{i,j\to j'} = a_{i,j\to j'}$. If both the provision and Y decreases ($a_{i,j\to j'} > 0$ and $b_{i,j\to j'} > 0$), we set $f_{i,j\to j'} = -b_{i,j\to j'}/a_{i,j\to j'}$. Finally, if the total provision increases ($a_{i,j\to j'} < 0$) or the total provision stays constant ($a_{i,j\to j'} = 0$) but Y decreases ($b_{i,j\to j'} > 0$), we set $f_{i,j\to j'} = -\infty$ indicating that this action will not be chosen. The update procedure is carried out iteratively, one action adjustment at a time, until the total provision is no greater than L as indicated in Eq. (3). In the case $\epsilon = 0$, there exists a theoretical bound to ensure that the optimal Y obtained from this procedure will not be less than half of the global optimal value [26].

IV. RESULTS AND DISCUSSIONS

A. Data set

We use the data set of a personal loan product provided by Kasikornbank. The data set consists of 600 loanees and five possible actions per loanee. Although we are made aware of which action is the DPO, due to data confidentiality we are not made aware of what the other four actions are. This does not, however, affect the model. $h_{i,j}$ is estimated from Q learning using the historical data of the loanees [27]. $l_{i,j}$ is related to historical data on repayments and default, loan-collection expenses, credit losses, economic conditions, interest rate, and tax policy [17]. Specific details of the derivations are omitted here due to data privacy. $A_{i,i'}$ is estimated from transactions between two loanees that happen internally within Kasikornbank. Note that the actual transactions $\tilde{A}_{i,i'}$ between the two loanees could be higher than this value, i.e., $\tilde{A}_{i,i'} = \alpha_{i,i'}A_{i,i'}$ with $\alpha_{i,i'} \geq 1$. However, it is not possible for a bank to collect every transaction from a customer. Therefore, we assume that $\alpha_{i,i'}$ does not depend on i and i', so that it can be absorbed into the redefinition of ϵ. The interpretation of ϵ in practice will be discussed later in the text.

The distributions of $h_{i,j}$, $l_{i,j}$, and $A_{i,i'}$ are depicted in Fig. 3. The units are made arbitrary for privacy concerns. The value of $h_{i,j}$ and $A_{i,i'}$ range from 0 to 0.7, while the value of $l_{i,j}$ ranges from approximately 10^{-5} to approximately 1. The financial network, as indicated by $A_{i,i'}$, is sparse with node degrees ranging from 0 to 7. We also benchmark our analysis with a simulated data set in Appendix A to ensure the generality of our results.

B. Numerical results

As we focus mainly on demonstrating the improvement on the yield Y and the provision as a proof of principle, we simulate the exact Hamiltonian evolution on a classical HPC cluster. The largest quantum system simulated consists of 35 qubits. To analyze the behavior of the QAOA, we compare our hybrid algorithm with a purely classical algorithm. The latter is achieved by directly feeding a random set of actions to the GPR algorithm. Typically,

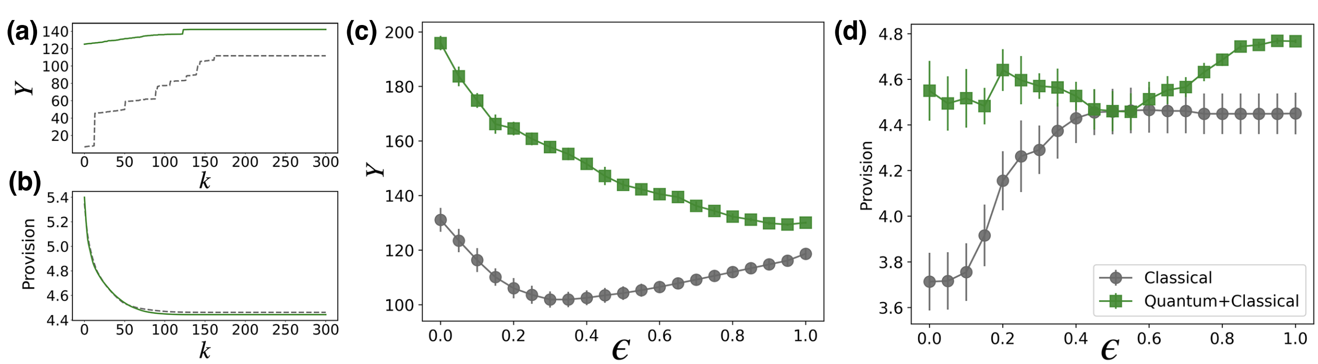

FIG. 4. Performance of the hybrid algorithm. (a),(b) shows Y and the total provision during the GPR algorithms with $\epsilon = 0.5$ as a function of the number of steps k in GPR, respectively. The solid green lines are when the solutions reconstructed from the QAOA are used as an input and the dashed gray lines are when random solutions are used as an input. (c),(d) shows Y and the total provision as a function of ϵ, respectively. The square green dots are from the hybrid algorithm and the circle gray dots are from the same optimization workflow but with the QAOA excluded. ($\nu = 7$, $T = 2$).

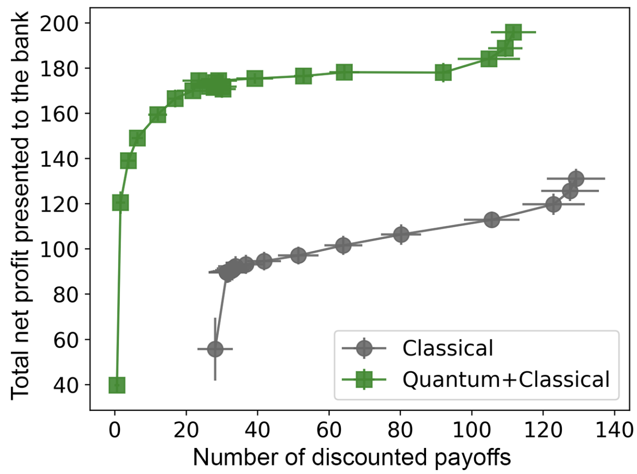

FIG. 5. Total net profit presented to the bank as a function of the number of DPO actions being taken. The square green dots are from the hybrid algorithm and the circle gray dots are from the standalone classical GPR where QAOA is removed from the optimization workflow. Note that our hybrid quantum-classical algorithm always yields a better total net profit presented to the bank compared to the standalone classical GPR algorithm ($\nu = 7, T = 2$).

multiple random initialization of a greedy optimizer is used to find a feasible solution for a nonconvex optimization problem by selecting the best outcome. Here, we look into the average yields and provisions from the solutions obtained from both algorithms to compare the effectiveness of the algorithms as a whole.

In Fig. 4(a), with $\epsilon = 0.5$ where the optimization objective values the expected return to the bank and the financial welfare among loanees equally, we see that the set of actions reconstructed from the QAOA starts with $Y \sim 120$ and then increases to ~ 140 during the GPR algorithm. On the other hand, a random set of actions start with $Y \sim 10$ and then increases to approximately 100, lower than the one with the QAOA. Figure 4(b) shows that, in both cases, the provision reduces from approximately 5.4 to approximately 4.4, resulting in 18.5% reduction. We find that for other values of ϵ, Y always monotonically increases, and the provision always monotonically decreases during the GPR algorithm. Hence, there is no need to introduce L to truncate the process. The minimum LLP is treated as a suggested LLP rather than a hard constraint.

Figure 4(c) shows Y after the GPR algorithm as a function of ϵ. We can see that the hybrid algorithm provides a higher Y compared to the standalone GPR algorithm for all values of ϵ. Figure 4(d) shows the provision after the GPR algorithm as a function of ϵ. At $\epsilon \lesssim 0.1$, the standalone GPR gives around 17% lower provision compared to the hybrid algorithm. This number is reduced to approximately 4.3% as ϵ goes towards unity. This result is expected as the QAOA attempts to increase Y alone without considering the LLPs. Nevertheless, the provision obtained from the hybrid algorithm is kept at approximately 4.6 for all values of ϵ. This value is 57.4% lower than the highest possible provision.

Finally, we note that ϵ may be hard to interpret in practice because it is normalized by the unknown variable $\alpha_{i,i'}$ as discussed above. To circumvent this, in Fig. 5, we plot the total net profit presented to the bank as a function of the optimal number of DPO actions N_f, i.e., the number of loanees with action $j = 1$. The latter is, in turn, varied by ϵ. Figure 5 provides an intuitive interpretation of the optimization results. The collector can choose the number of DPO actions based on their experience, then read out the estimated net profit that will be made. We find that, for the same N_f, the hybrid algorithm always gives a higher expected return. Specifically, for $N_f \sim 30$–100, the expected profit from the hybrid algorithm is approximately 70% higher than the standalone GPR. In addition, with the hybrid algorithm, we find that the expected return shows a plateau at $N_f \sim 20$–100. This implies that the collector may choose to apply, say, 30 DPOs to get the profit of approximately 175, instead of 120 DPOs (300% more) to get a profit of approximately 200 (only 14.3% more).

V. CONCLUSIONS

We devise a hybrid quantum-classical algorithm to solve loan-collection problems with LLPs by formulating them as QCBO models. Our approach allows the lender to maximize the expected net return while accounting for LLPs

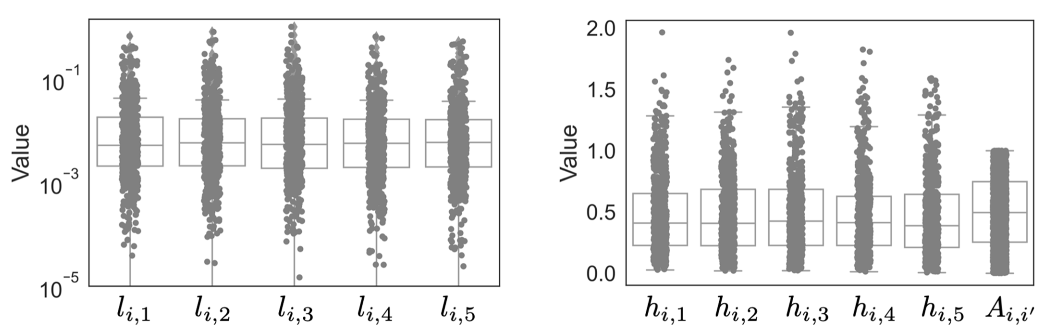

FIG. 6. Statistics of model parameters in the simulated data set. (Left) Box plot of $l_{i,j}$ in a log scale. (Right) Box plot of $h_{i,j}$ and $A_{i,i'}$ showing their distributions and quantiles.

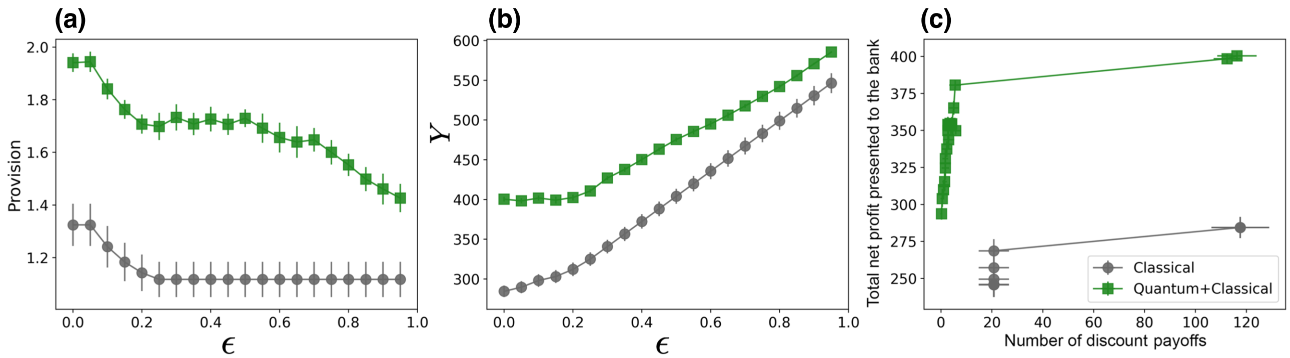

FIG. 7. Performance of the hybrid algorithm on the simulated data set. (a),(b) shows the total provision and the yield function Y as a function of ϵ, respectively. The square green dots are from the hybrid algorithm and the circle gray dots are from the standalone GPR. (c) Total net profit presented to the bank as a function of the number of DPO actions being taken. The square green dots are from the hybrid algorithm and the circle gray dots are from the standalone classical GPR. ($\nu = 7$, $T = 2$.)

and the financial well being of the loanees, as measured by the association matrix. Compared to a purely classical approach, our hybrid algorithm provides a higher net profit, regardless of the amount of the network effect as measured by ϵ. The algorithm also suggests the collector with the lowest number of DPO actions that still gives a relatively high expected return. Our work is not a proof of quantum advantage. Rather it demonstrates a practical use case for quantum technology that warrants future study.

---

input : (i) Association matrix $A = (A_{i,i'})$.
(ii) Maximum number of nodes per subgraphs $\nu \in \mathbb{Z}^+$.
output: A set of subgraphs. Each subgraph represents a group of loanees with high association among them. Each node in each subgraph represents one loanee.
begin
  $\mathcal{S} \leftarrow \{\}$
  $G \leftarrow \langle$a graph obtained from A $\rangle$
  $\mathcal{S}^{(1)} \leftarrow \langle$a set of subgraphs of G obtained from greedy modularity$\rangle$
  for $g^{(1)} \in \mathcal{S}^{(1)}$ **do**
    $\mathcal{S} \leftarrow \mathcal{S} \cup$ `RecLouvain`$(g^{(1)})$
  return $\mathcal{S}$

function `RecLouvain`$(g^{(1)})$
  $\mathcal{S}^{(3)} \leftarrow \{\}$
  $\mathcal{S}^{(2)} \leftarrow \langle$a set of subgraphs of $g^{(1)}$ obtained from the Louvain method$\rangle$
  for $g^{(2)} \in \mathcal{S}^{(2)}$ **do**
    % Introduce edge nodes
    $\mathcal{W} \leftarrow []$
    $\mathcal{V} \leftarrow \langle$a set of nodes in $g^{(2)}\rangle$
    for $i \in \mathcal{V}$ **do**
      $\mathcal{E}_i \leftarrow \langle$a set of edges in G that are incident to $i\rangle$
      for $(i, i') \in \mathcal{E}_i$ **do**
        if $i' \notin \mathcal{V}$ and $i' \notin \mathcal{W}$ **then**
          $\mathcal{W} \leftarrow \mathcal{W} \cup \{i'\}$
    $g^{(2)} \leftarrow \langle$add nodes in $\mathcal{W}$ and corresponding edges to $g^{(2)}\rangle$
    % Recursive division
    if $|g^{(2)}| > \nu$ **then**
      $\mathcal{S}^{(3)} \leftarrow \mathcal{S}^{(3)} \cup$ `RecLouvain`$(g^{(2)})$
    else
      $\mathcal{S}^{(3)} \leftarrow \mathcal{S}^{(3)} \cup \{g^{(2)}\}$
  return $\mathcal{S}^{(3)}$

Algorithm 1. Recursive division

input : (i) Local fields $\{h_{i,j}\}$ for subgraph g.
(ii) Association $\{A_{i,i'}\}$ for subgraph g.
(iii) The number of driving cycles T in QAOA.
(iv) The maximum number of iterations η in the COBYLA.
output: A set of probabilities $\{p(\underline{x})|\forall\underline{x}\in\{0,1\}^{N_g\times M}\}$, where N_g is the number of loanees in subgroup g. Configurations $\underline{x}$ that have high values of the objective function will have high $p(\underline{x})$.
begin
- $\underline{\theta}\leftarrow\langle$a vector of size $2T$ containing randomized real numbers $\in[0,1)\rangle$
- $\underline{\theta}_{\rm opt}\leftarrow\langle$minimizes $\texttt{Energy}(\underline{\theta})$ using COBYLA$\rangle$
- $|\psi_{\rm opt}\rangle\leftarrow\texttt{Evolve}(|\psi_0\rangle,\underline{\theta}_{\rm opt})$
- **return** $\{p(\underline{x})=|\langle\underline{x}|\psi_{\rm opt}\rangle|^2\}$

function $\texttt{Energy}(\underline{\theta})$
- $|\psi\rangle\leftarrow\texttt{Evolve}(|\psi_0\rangle,\underline{\theta})$
- **return** $\langle\psi|\hat{H}_B|\psi\rangle$

function $\texttt{Evolve}(|\psi\rangle,\underline{\theta})$
- **for** $t=0$ to $T-1$ **do**
  - % Exploitation
  - $\hat{U}_A\leftarrow\exp\left(-i\hat{H}_A\cdot\underline{\theta}[2t]\right)$
  - % Exploration
  - $\hat{U}_B\leftarrow\exp\left(-i\hat{H}_B\cdot\underline{\theta}[2t+1]\right)$
  - $|\psi\rangle\leftarrow\hat{U}_A\hat{U}_B|\psi\rangle$
- **return** $|\psi\rangle$

Algorithm 2. QAOA with local constraints

DATA AVAILABILITY

The supporting data is available upon a reasonable request to corresponding authors.

ACKNOWLEDGMENTS

We thank Vorapong Suppakitpaisarn for fruitful discussions. P. Palittapongarnpim, P. Chaiwongkhot, and

input : (i) A set of the subgraphs $\mathcal{S}$ obtained from the Recursive Division algorithm.
(ii) A set of probabilities for every subgraphs $\{p_g(\underline{x})|\forall\underline{x}\in\{0,1\}^{N_g\times M},\forall g\in\mathcal{S}\}$ obtained from QAOA.
(iii) A hyper-parameter for the maximum number of candidates per subgraph λ.
output: A set of actions that maximises Y.
begin
- $g_L\leftarrow\langle$a randomly chosen subgraph from $\mathcal{S}\rangle$
- $\mathcal{X}_L\leftarrow\langle$a set of the first λ bit-strings $\{\underline{x}\}$ from subgraph g_L that have the highest probabilities $\{p_{g_L}(\underline{x})\}\rangle$
- **for** $g_R\in\mathcal{S}\setminus\{g_L\}$ **do**
  - $\mathcal{X}_R\leftarrow\langle$a set of the first λ bit-strings $\{\underline{x}\}$ from subgraph g_R that have the highest probabilities $\{p_{g_R}(\underline{x})\}\rangle$
  - $\mathcal{W}_{LR}\leftarrow\langle$a set of edge nodes between g_L and $g_R\rangle$
  - $\mathcal{X}_{LR}=\{\}$
  - **for** $\underline{x}_L\in\mathcal{X}_L$ **do**
    - **for** $\underline{x}_R\in\mathcal{X}_R$ **do**
      - $\mathcal{X}_{LR}\leftarrow\mathcal{X}_{LR}\cup\{\texttt{Combine}(\underline{x}_L,\underline{x}_R,\mathcal{W}_{LR})\}$
  - **while** $|\mathcal{X}_{LR}|==0$ **do**
    - $\underline{x}'_R\leftarrow\langle$the next-highest-probability bit-string from $g_R\rangle$
    - **for** $\underline{x}_L\in\mathcal{X}_L$ **do**
      - $\mathcal{X}_{LR}\leftarrow\mathcal{X}_{LR}\cup\{\texttt{Combine}(\underline{x}_L,\underline{x}'_R,\mathcal{W}_{LR})\}$
  - **if** $|\mathcal{X}_{LR}|>\lambda$ **then**
    - $\mathcal{X}_{LR}\leftarrow\langle$only keep the first λ elements that have the highest $Y\rangle$
  - $g_L\leftarrow\langle$combine nodes and edges in g_L and g_R.$\rangle$
  - $\mathcal{X}_L\leftarrow\mathcal{X}_{LR}$
- **return** $\mathcal{X}_L$

function $\texttt{Combine}(\underline{x}_L,\underline{x}_R,\mathcal{W}_{LR})$
- **if** actions at the edge nodes from $\underline{x}_L$ and $\underline{x}_R$ are the same **then**
  - **return** $\langle$a combined bit-string $\underline{x}_{LR}$ as explained in the main text.$\rangle$

Algorithm 3. State reconstruction

```
input  : (i) A set of actions j, where the element j[i] ∈ {1, 2, .., M} is the optimal action j that is taken to loanee i.
         (ii) The maximum number of iteration η.
output: A new optimal actions that minimises the LLP.
begin
    j_best ← j
    y_best ← ⟨The objective function for j⟩
    l_best ← ⟨The LLP of j⟩
    for k = 1 to k = η do
        F ← {}
        for i = 1 to i = N do
            for j' = 1 to j' = M do
                j_new ←⟨a new set of actions where j[i] is changed to j'⟩
                y_new ←⟨the objective function for j_new⟩
                l_new ←⟨the LLP of j_new⟩
                % Reward
                a_{i,j→j'} ← l_best − l_new
                % Penalty
                b_{i,j→j'} ← y_best − y_new
                % Finesse score
                if a_{i,j→j'} > 0 and b_{i,j→j'} ≤ 0 then
                    f_{i,j→j'} ← a_{i,j→j'}
                if a_{i,j→j'} > 0 and b_{i,j→j'} > 0 then
                    f_{i,j→j'} ← −b_{i,j→j'}/a_{i,j→j'}
                if a_{i,j→j'} < 0 then
                    f_{i,j→j'} ← −∞
                if a_{i,j→j'} = 0 and b_{i,j→j'} > 0 then
                    f_{i,j→j'} ← −∞
                F ← F ∪ {f_{i,j→j'}}
        j_best ←⟨a new set of actions where j[i] is changed to j' such that f_{i,j→j'} ∈ F is maximized⟩
        y_best ← ⟨The objective function for j⟩
        l_best ← ⟨The LLP for j⟩
    return u_best
```

Algorithm 4. Greedy Provision Reduction

T. Chotibut are supported by the Program Management Unit for Human Resources and Institutional Development, Research and Innovation (Grant No. B05F650024). T. Chotibut is also supported by Thailand Science Research and Innovation Fund Chulalongkorn University [CU_FRB65_ind (5)_110_23_40]. J. Tangpanitanon, P. Prugsanapan, N. Raksasri, W. Wannasiwaporn, P. Thajchayapong, and Y. Raksri are supported by Kasikorn Business-Technology Group, Bangkok, Thailand. Lastly, we acknowledge the National Science and Technology Development Agency, National e-Science Infrastructure Consortium, Chulalongkorn University and the Chulalongkorn Academic Advancement into Its 2nd Century Project (Thailand) for providing computing infrastructure that has contributed to the research results reported within this paper (www.e-science.in.th).

APPENDIX: BENCHMARKING WITH A SIMULATED DATA SET

In this section, we apply the algorithms presented in the main text to a simulated data set to show the generality of our results. We use $M = 5$ and $N = 600$ as in the main text. To investigate the performance of our algorithm on other realistic financial networks, we model the interconnectivity of assets based on the observation of Austrian financial transaction network [28]. In particular, the undirected interconnectivity from daily financial transactions among Austrian assets is effectively captured by the Erdős-Renyi ensemble with the mean degree 5.78, whereas the weights are rather independent from the network topology. Accordingly, we draw our simulated network interconnectivity from the Erdős-Renyi ensemble with mean degree 4 and $N = 600$ nodes, with the association matrix element (weight) $A_{i,i'}$ sampled independently from the uniform distribution between $[0, 1]$. As $V_i \equiv \sum_{i'} A_{i,i'}$ should effectively represent the value of the asset i, we assume that the maximum net return the bank can attain from a loan-collection action of the asset i is the fraction f of V_i, i.e., $h_i^{\max} \equiv \max_j \{h_{i,j}\} = fV_i$. Specifically, we set $f = 0.2$ in our simulated data set. To account for the heterogeneity of $h_{i,j}$, we model the expected return attained by the bank from each of the five collection actions to be $h_i^{\max} \times (1, 0.8, 0.6, 0.4, 0.2)$ in the order that is randomly shuffled for each loanee i. $l_{i,j}$'s are computed from Kasikornbank's data set in a similar fashion to the main text, which are normalized to make the unit arbitrary for privacy concern. The statistics of the simulated data set are shown in Fig. 6.

In Fig. 7(c), we plot the total net profit present to the bank as the number of DPO actions using optimum solutions from the hybrid quantum-classical and the standalone classical algorithm as in the main text. We find that the hybrid algorithm gives a higher return compared to the classical counterpart as expected. The total provision as well as Y as a function of ϵ are shown in Figs. 7(a) and 7(b), respectively. The yield function enjoys more attractive optimization results from the hybrid algorithm. Although the provision from the hybrid algorithm is larger than that of the classical one (for the same reason explained in the main text), it is kept at a maximum of approximately 2, which is 75% lower than the largest possible total provision.

APPENDIX B . PSEUDOCODES FOR HYBRID QUANTUM-CLASSICAL ALGORITHMS

In this section, we provide below pseudocodes for four algorithms involved in the hybrid quantum-classical algorithms. These are (i) recursive division in the divide-and-conquer algorithm, (ii) QAOA with local constraints, (iii) state reconstruction, which is the second part of the divide-and-conquer algorithm, and (iv) greedy provision reduction. See the main text for an overview discussion for each algorithm.

[1] R. Orús, S. Mugel, and E. Lizaso, Quantum computing for finance: Overview and prospects, Rev. Phys. **4**, 100028 (2019).

[2] D. J. Egger, C. Gambella, J. Marecek, S. McFaddin, M. Mevissen, R. Raymond, A. Simonetto, S. Woerner, and E. Yndurain, Quantum computing for finance: State of the art and future prospects, IEEE Trans. Quantum Eng. **1**, 1 (2020).

[3] A. Bouland, W. van Dam, H. Joorati, I. Kerenidis, and A. Prakash, arXiv:2011.06492.

[4] M. Hodson, B. Ruck, H. Ong, D. Garvin, and S. Dulman, Portfolio rebalancing experiments using the quantum alternating operator ansatz (2019). arXiv:1911.05296.

[5] J. Cohen, A. Khan, and C. Alexander, Portfolio optimization of 40 stocks using the dwave quantum annealer (2020). arXiv:2007.01430.

[6] S. Mugel, M. Abad, M. Bermejo, J. Sánchez, E. Lizaso, and R. Orús, Hybrid quantum investment optimization with minimal holding period, Sci. Rep. **11**, 19587 (2021).

[7] D. Venturelli and A. Kondratyev, Reverse quantum annealing approach to portfolio optimization problems, Quantum Mach. Intell. **1**, 17 (2019).

[8] N. Stamatopoulos, D. J. Egger, Y. Sun, C. Zoufal, R. Iten, N. Shen, and S. Woerner, Option pricing using quantum computers, Quantum **4**, 291 (2020).

[9] S. Ramos-Calderer, A. Pérez-Salinas, D. García-Martín, C. Bravo-Prieto, J. Cortada, J. Planagumà, and J. I. Latorre, Quantum unary approach to option pricing, Phys. Rev. A **103**, 032414 (2021).

[10] F. Fontanela, A. Jacquier, and M. Oumgari, Short Communication: A Quantum Algorithm for Linear PDEs Arising in Finance, SIAM J. Financ. Math. **12**, SC98 (2021).

[11] M. C. Braun, T. Decker, N. Hegemann, S. F. Kerstan, and C. Schäfer, arXiv:2103.05475.

[12] J. Alcazar, A. Cadarso, A. Katabarwa, M. Mauri, B. Peropadre, G. Wang, and Y. Cao, Quantum algorithm for credit valuation adjustments, New J. Phys. **24**, 023036 (2022).

[13] 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:2201.02773 (2022).

[14] J. P. Harding and C. Sirmans, Renegotiation of troubled debt: The choice between discounted payoff and maturity extension, Real Estate Econ. **30**, 475 (2002).

[15] M. Bardoscia, P. Barucca, S. Battiston, F. Caccioli, G. Cimini, D. Garlaschelli, F. Saracco, T. Squartini, and G. Caldarelli, The physics of financial networks, Nat. Rev. Phys. **3**, 490 (2021).

[16] J. Nin, B. Salbanya, P. Fleurquin, E. Tomás, A. Arenas, and J. J. Ramasco, Modeling financial distress propagation on customer–supplier networks, Chaos **31**, 053119 (2021).

[17] P. K. Ozili and E. Outa, Bank loan loss provisions research: A review, Borsa Istanbul Rev. **17**, 144 (2017).

[18] D. Kiarelly, G. de Araujo, B. H. Cohen, and P. Pogliani, Bank loan loss provisioning during the Covid crisis, BIS Q. Rev. **March 2021** (2021).

[19] D. M. Abrams, N. Didier, B. R. Johnson, M. P. da Silva, and C. A. Ryan, Implementation of XY entangling gates with a single calibrated pulse, Nat. Electron. **3**, 744 (2020).

[20] A. Shoghi, Debt collection industry: Machine learning approach, J. Money Econ. **14**, 453 (2019).

[21] Z. Liu, P. He, and B. Chen, A Markov decision model for consumer term-loan collections, Rev. Quant. Finan. Account. **52**, 1043 (2019).

[22] J. Li, M. Alam, and S. Ghosh, Large-scale quantum approximate optimization via divide-and-conquer, IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **1**, 1 (2022).

[23] 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 **12**, 34 (2019).

[24] A. Clauset, M. E. J. Newman, and C. Moore, Finding community structure in very large networks, Phys. Rev. E **70**, 066111 (2004).

[25] A. Lancichinetti and S. Fortunato, Community detection algorithms: A comparative analysis, Phys. Rev. E **80**, 056117 (2009).

[26] J. Kleinberg and E. Tardos, *Algorithm Design: Pearson New International Edition* (Pearson, Boston, 2013), Chap. *11.8*.

[27] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction* (The MIT Press, Cambridge, Massachusetts, 2018), 2nd ed.

[28] F. Kyriakopoulos, S. Thurner, C. Puhr, and S. W. Schmitz, Network and eigenvalue analysis of financial transaction networks, Eur. Phys. J. B **71**, 523 (2009).