Management Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Improving the Efficiency of Payments Systems Using Quantum Computing

Christopher McMahon, Donald McGillivray, Ajit Desai, Francisco Rivadeneyra, Jean-Paul Lam, Thomas Lo, Danica Marsden, Vladimir Skavysh

informs
https://pubsonline.informs.org/journal/mnsc

MANAGEMENT SCIENCE
Vol. 70, No. 10, October 2024, pp. 7325–7341
ISSN 0025-1909 (print), ISSN 1526-5501 (online)

Improving the Efficiency of Payments Systems Using Quantum Computing

Christopher McMahon,[a] Donald McGillivray,[a] Ajit Desai,[b,*] Francisco Rivadeneyra,[b] Jean-Paul Lam,[a,c] Thomas Lo,[a] Danica Marsden,[b] Vladimir Skavysh[b]

[a] GoodLabs Studio, Toronto, Ontario M5H 3E5, Canada; [b] Banking and Payments, Bank of Canada, Ottawa, Ontario K1A 0G9, Canada; [c] Department of Economics, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada

*Corresponding author

Contact: cmcmahon@goodlabs.studio, https://orcid.org/0000-0001-7802-3016 (CM); dmcgillivray@goodlabs.studio (DM); adesai@bankofcanada.ca, https://orcid.org/0000-0001-7098-3186 (AD); frivadeneyra@bankofcanada.ca (FR); jplam@uwaterloo.ca, https://orcid.org/0000-0002-7131-8636 (J-PL); tlo@goodlabs.studio (TL); dmarsden@bankofcanada.ca (DM); vskavysh@bankofcanada.ca (VS)

Received: February 2, 2023
Revised: November 10, 2023
Accepted: February 7, 2024
Published Online in Articles in Advance: July 12, 2024

https://doi.org/10.1287/mnsc.2023.00314

Abstract. High-value payment systems (HVPSs) are typically liquidity intensive because payments are settled on a gross basis. State-of-the-art solutions to this problem include algorithms that seek netting sets and allow for ad hoc reordering of submitted payments. This paper introduces a new algorithm that explores the entire space of payments reordering to improve the liquidity efficiency of these systems without significantly increasing payment delays. Finding the optimal payment order among the entire space of reorderings is, however, an NP-hard combinatorial optimization problem. We solve this problem using a hybrid quantum annealing algorithm. Despite the limitations in size and speed of today's quantum computers, our algorithm provides quantifiable liquidity savings when applied to the Canadian HVPS using a 30-day sample of transaction data. By reordering batches of 70 payments, we achieve an average of Canadian (C) $240 million in daily liquidity savings, with a settlement delay of approximately 90 seconds. For a few days in the sample, the liquidity savings exceed C$1 billion. Compared with classical computing and with current algorithms in HVPS, our quantum algorithm offers larger liquidity savings, and it offers more reliable and consistent solutions, particularly under time constraints.

History: Accepted by Chung Piaw Teo, optimization.

Supplemental Material: The data files are available at https://doi.org/10.1287/mnsc.2023.00314.

Keywords: quantum algorithm • combinatorial optimization • NP hard • high-value payments system

1. Introduction

High-value payment systems (HVPSs), used to settle transactions between large financial institutions, are part of the core financial infrastructure of every country. Central banks that operate or oversee these systems are mandated to ensure their safety and efficiency. Most HVPSs around the world—including the United States's Fedwire, the Eurosystem's TARGET2, and Canada's Lynx—are real-time gross settlement (RTGS) systems that settle each payment request on an individual basis. These systems are liquidity intensive; participants are required to have liquidity before payments are processed.[1] Liquidity has an opportunity cost, incentivizing participants to wait for incoming payments before making their own. This behavior can reduce their efficiency, causing delays in payments and even gridlock (Bech and Garratt 2003).

Given the systemic importance of HVPSs, improving their efficiency is an important area of research for central banks and academics alike (Martin and McAndrews 2008b, Rivadeneyra and Zhang 2020, Alexandrova-Kabadjova et al. 2023). Improving the liquidity efficiency of an HVPS can be done in two main ways. The first is to incentivize participants to increase the coordination of the timing of their payment submissions. The second is, once payments are submitted to the system, to change the order in which payments will be settled to find a sequence of settlement with lower liquidity demand or a netting set. The state-of-the-art solutions to the liquidity efficiency problem, called liquidity-saving mechanisms (LSMs), involve algorithms that seek netting sets and some ad hoc and limited reordering of the payments submitted to the system.[2]

In this paper, we propose the full use of reordering of payments to improve the efficiency of HVPS and the use of quantum computing techniques to solve the reordering problem. Reordering can increase the efficiency of HVPS through its effect on liquidity recycling.[3] Reordering, however, has not been fully explored as a solution given its computational complexity. Payments reordering is an NP-hard combinatorial optimization problem that grows very rapidly with the number of payments in a batch ($O(N!)$) with a time complexity similar to that of the well-known traveling salesman problem (Jünger et al. 1995). For the first time, quantum techniques, although not expected to be able to solve such problems in polynomial time, offer significant speedups at meaningful scales for the reordering problem.

More specifically, we propose an algorithm whose objective is to search over the entire space of orderings of a batch of payments to reduce the liquidity usage in that batch without significantly increasing the average delay of settlement. From an economic perspective, the proposed algorithm can be viewed as a centralized payment preprocessor. In this setup, participants submit their payment requests to this algorithm before submitting them to the actual system. Subsequently, after the optimization using a quantum computing or classical solver, our algorithm provides the participants with a recommended payment order. This reordered sequence minimizes the total liquidity needed for settling the payments compared with settling them in their original submission order. Although participants could effectively veto the proposal by not submitting their payments in the suggested order, in practice, if the suggested reorderings almost always reduce aggregate liquidity demands, participants are unlikely to do so strategically.

For combinatorial optimization problems such as this, quantum annealers have advanced to a point where it is worth investigating their capabilities against classical-only approaches, especially when they are combined with classical computing in a hybrid algorithm (Parekh et al. 2016, National Academies of Sciences 2019, Egger et al. 2020). Quantum annealing leverages quantum superposition and tunneling to effectively explore the energy landscape of optimization problems, potentially locating global minima faster than classical methods (Kadowaki and Nishimori 1998). Additionally, unlike universal quantum computers, which require error correction and complex gate operations to perform computations, quantum annealers have simpler architecture and are effective in tackling certain optimization problems (Finnila et al. 1994, Farhi et al. 2001).

We formulate the reordering of a batch of payments in an HVPS as a mixed binary optimization (MBO). This MBO is then transformed into a quadratic unconstrained binary optimization (QUBO), which can then be easily converted into an Ising model suitable for execution on a quantum annealer. Given the scale of our problem and the current topology of quantum annealers in terms of their qubit count and connectivity, we reframe the final problem as a constrained quadratic model (CQM). We optimize it on D-Wave's quantum annealer, leveraging their hybrid quantum solver explicitly designed for CQMs (D-Wave Systems 2021). To ensure a comprehensive benchmarking exercise, we employ the SCIP optimization suite—a classical algorithm acclaimed for mixed integer programming (Bestuzheva et al. 2021, Rehfeldt et al. 2022).

We test the payments reordering algorithm, optimized with the CQM and SCIP solvers, on a sample of 30 days of nonurgent transactions from the Canadian HVPS. Each day, approximately 23,000 payments are divided into batches of 70 payments, which are then fed into the solvers. The CQM solver, taking just five seconds of solve time, finds liquidity savings in 26% of the batches compared with the original order (first in, first out (FIFO)). This results in an average liquidity savings of C$240.8 million, which is about 1.4% of the typical liquidity usage. On some days, savings exceed C$1 billion, with the maximum savings reaching up to 7% of usage. We also observe that the total liquidity saved for each participant is proportional to the participants' total incoming and outgoing transaction values. When then compare the reordering solution provided by the CQM with that of SCIP under the same solution time constraint (five seconds). On average, SCIP is able to provide C$213.9 million end-of-day savings over FIFO, representing 87% of the savings achieved with the CQM reordering.

These results show that reordering algorithms have the potential to improve the efficiency of HVPS, solved with either quantum or classical techniques. The obvious question is then as follows. Is quantum necessary? Although the classical solver was quite effective at solving the reordering problem at this scale, quantum showed several advantages. First, the quantum solver proved more consistent at providing a solution under five seconds and for those solutions to provide larger liquidity savings. CQM always provided a solution under five seconds, whereas SCIP did so for 93% of the batches. Although SCIP found solutions comparable with the CQM in most batches, it performed worse than the CQM in many cases and only slightly better in a few.

The last aspect where quantum significantly outperformed classical techniques was with larger batches. With a batch size of 140 payments, the CQM reordering resulted in greater savings. We observed that over 60% of the batches were optimizable, leading to end-of-day savings increasing by 94% and 326% compared with a batch size of 70 on the two days in our sample that we were able to test. In contrast, SCIP only achieved 38%

and 18% of the savings attained by the CQM on those two days. These findings indicate that liquidity savings have the potential to significantly increase with the batch size because the scope of liquidity recycling increases accordingly. As quantum computing capabilities are rapidly advancing, we expect that larger reordering problems will be tackled more effectively with quantum than with classical techniques.

We also perform a comparison of our quantum solution of the reordering problem with traditional LSMs in use today. We use the Canadian HVPS, called Lynx, as comparison. Under the same assumption of batch settlement with a fixed delay, the CQM reordering compared quite favorably. Although the Canadian HVPS combines several LSMs—including bypass and the more advantageous bilateral and multilateral netting—the CQM provided slightly larger liquidity savings.

This paper provides two main messages. First, reordering is a promising avenue for improving the efficiency of HVPS and possibly other financial infrastructures relying on gross settlement. Second, quantum annealing is a useful technique to solve reordering problems, particularly when the business application imposes a constraint on the solve time. Our solution method suggests ways to test quantum annealing speedups of reordering problems over classical solvers.

2. Literature Review

The problem of liquidity intensity of RTGS systems is well known, and alternatives have been explored, with LSMs being a prominent solution (Martin and McAndrews 2008b, Diehl and Schollmeyer 2009, Atalay et al. 2010, Davey and Gray 2014). The purpose of LSMs is to reduce the liquidity requirements of an HVPS without the credit risks associated with delay net settlement models.[4] Typically, LSMs involve the use of algorithms that seek netting sets and some ad hoc reordering, like bypassing larger payments submitted to the system. Because of their simplicity and practicality, these LSMs have been widely adopted worldwide. The efficiency improvements of these LSMs theoretically depend on the liquidity regime (Martin and McAndrews 2008a, b; Jurgilas and Martin 2013). These algorithms have been used extensively around the world with some success. For the UK system, Davey and Gray (2014) estimated a reduction in liquidity requirements of 20%. Some recent empirical evidence, however, is casting doubt on their effectiveness in enhancing the efficiency of HVPS (Alexandrova-Kabadjova et al. 2023).

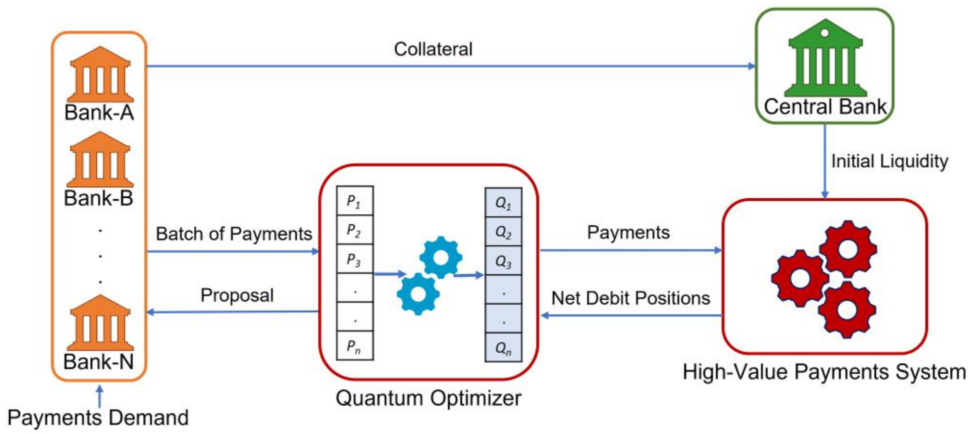
In practice, LSMs are designed based on heuristics, and their effectiveness is assessed through simulation methods (Galbiati and Soramaki 2010; Rivadeneyra and Zhang 2020, 2022). However, because of computational constraints, the algorithms employed thus far have not been engineered to explore the entire solution space. In contrast, our approach, enabled by quantum computing, attempts to evaluate the entire space of reordering looking for liquidity savings. This opens up new avenues for the exploration and design of future LSMs, which could also be utilized alongside existing LSMs. Recently, Garratt (2022) introduced an alternative approach based on the use of the Shapley value cost allocation method. In this approach, participants are presented with incentive-compatible take-it-or-leave-it offers to provide the necessary liquidity, ensuring that welfare-maximizing netting proposals are consistently accepted. However, this approach also comes with challenges, such as the necessity of knowing the individual valuations of settling each payment and the computational intensity of calculating Shapley values for large sets of payments.

Quantum computing technology is evolving rapidly, and there is growing interest in its potential for solving certain classes of problems, such as optimization, faster than classical computers (National Academies of Sciences 2019, Jünger et al. 2021, Tasseff et al. 2022). Using the quantum properties of superposition, entanglement, and tunneling, quantum computers can search large-input parameter spaces faster than any classical computer could even theoretically achieve (Steane 1998, Hidary 2019). Researchers have begun exploring using quantum computing to solve various problems arising in economics—including queuing problems that arise in financial markets (Woerner and Egger 2019, Egger et al. 2020, Hull et al. 2020, Stamatopoulos et al. 2020, Fernández-Villaverde and Hull 2023, Skavysh et al. 2023). Closely related to our application, Braine et al. (2021) extend an algorithm for an MBO problem applied to transaction settlement. They optimize small batches of securities trading transactions using gate-based quantum computing. However, their optimization focuses on determining the number of transactions that can be settled while allowing for netting when the latter does not apply to RTGS systems.

3. High-Value Payment Systems

In an HVPS, the participating institutions process the payment requests received from their clients. To do so, they apportion collateral to the central bank in exchange for the liquidity necessary to settle those payment requests. The system settles the payments in the order they arrive to the system if they satisfy risk controls. The efficiency of these systems is determined by the amount of liquidity that the participants choose to allocate to the system and by the timing of their payment requests. Participants can delay submitting their payments to await for incoming payments to fund their own payments, reducing the need for their initial liquidity allocation. Such incentives create a trade-off

Figure 1. (Color online) Schematic of the Quantum Optimizer as a Preprocessor to the Payment System



Notes. There are N participating banks; they receive multiple payment requests from their clients throughout the day. Banks submit payments to the central queue in the order $\{P_1, P_2, \ldots, P_n\}$. The quantum optimizer then processes those payments and proposes a new order $\{Q_1, Q_2, \ldots, Q_n\}$. If each bank accepts the proposal, those payments will be submitted to the HVPS. Participant's liquidity positions are provided to the quantum optimizer by the system.

between liquidity and delay (Bech and Garratt 2003, Castro et al. 2020). Our reordering algorithm seeks to find better solutions to the liquidity management problem by optimizing the order of payments without significantly increasing the settlement delay.

Figure 1 is a stylized schematic of a wholesale payment system. The figure also shows a generic quantum optimizer as a central preprocessing mechanism between participants and the payment system. A key challenge for new algorithms seeking to improve the efficiency of existing payments systems is how to incorporate them without incurring a fundamental change to the system. As a preprocessor, quantum optimization algorithms could be used to propose orderings that tangibly benefit the participants, incentivizing them to submit the payments to the system in the order suggested by the algorithm.

In our approach, each participant submits payment requests to a centralized queue denoted in the ordered set $\{P_1, P_2, \ldots, P_n\}$. Each element in this set contains information about the value of the payment, the two participants involved (payer and payee), and the time the request was submitted. Once there are enough payments in the queue (a batch), the quantum optimizer processes those payments and provides a proposal for an optimized queue in the order $\{Q_1, Q_2, \ldots, Q_n\}$ that minimizes liquidity requirements and contains the new submission timing. If all banks accept the proposal, those payments are submitted to the system in the new order proposed by the quantum optimizer. As payments are settled throughout the day, each participant's liquidity position (balance) changes; this information—necessary for the optimization process—is fed to the quantum optimizer by the payment system.

Our approach is to have a fixed size n of the batch of payments that will be optimized by the quantum optimizer. Once the number of payments received reaches that size, the optimization is triggered. Because the flow of payment requests varies through the day, the optimization occurs with varied frequency (see Figure 3). To evaluate the reordering algorithm, we report delay statistics that consider the time needed to collect enough payments (wait time) for the batch and the time required by the combined compute resources—a hybrid of quantum and classical—to evaluate the optimized queue (processing time). Given the wait time, we choose to test the reordering algorithm on nonurgent payments.

4. Quantum Formulation

The goal of the optimization is to minimize the aggregate liquidity needed in the system to settle a given set of payments, ordered with index $i = 1 \ldots n$. To measure the liquidity used in the system, we start by defining $N(i)$ as a participant's net position before a payment i is settled. A participant's net position is the liquidity balance accounting for all incoming and outgoing payments up to that time in the day.

Next, we define mNDP as the maximum net debit position experienced by a participant at any time thus far in the day. If a participant begins the day with $N = \$0$, then the end-of-day mNDP after all payments have settled measures the minimum liquidity a participant would have required to process its payments without incurring a negative position. We can, therefore, think of mNDP as equivalent to a participant's minimum liquidity requirements. Reducing the sum of mNDP across all participants, holding payment values and settlement delay fixed, represents an improvement in the liquidity efficiency of the system.

The quantum optimizer preprocessor rearranges the queue of payment instructions with an alternative index $t = 1 \ldots n$ that meets the goal of minimizing total liquidity needs subject to the constraints that no participant gets into a negative liquidity position at any point in time and that no payment remains in the queue.

4.1. Solvers: D-Wave-CQM and SCIP

The state-of-the-art tools in mathematical programming for addressing complex problems are tools such as SCIP, CPLEX, and Gurobi. SCIP, however, stands out for many reasons (Bestuzheva et al. 2021). SCIP is open source and under continuous development, reflecting the field's current state, and it has undergone rigorous benchmarking and comparison with other solvers (Anand et al. 2017, Vigerske and Gleixner 2018, Rehfeldt et al. 2022, Şeker et al. 2022, Bestuzheva et al. 2023, Mittelmann 2023). We can infer potential outcomes of quantum methods in real-world applications by comparing quantum annealing with SCIP's performance on these problems. SCIP uses branch-and-bound search to locate the exact optimal solution. It makes use of many techniques to aid in its speed and efficiency while searching the tree of solutions, such as problem reduction, presolving, and decomposition (see details in Appendix A).

An added complexity to the comparison between solvers is the hardware aspect. Classical optimization methods have been tested and standardized across a broad spectrum of hardware from personal computers to high-performance clusters. In contrast, quantum annealing and other quantum computing methods depend on specialized, cutting-edge hardware. Neither hardware platform is capable of running the other algorithm, making it challenging to create standardized benchmarks that fairly compare the two approaches. Thus, in order to compare the performance of the CQM hybrid solver with a classical solver, we utilized the SCIP software package deployed on an AWS t2.xlarge computing instance, which is equipped with four 3.0-GHz Intel Scalable Processors and 16 GB of memory. It is sufficiently powerful so as to provide a robust benchmark, but it is also reasonably accessible.

A variety of different technologies are being explored as the best means to build quantum hardware, but they are primarily classified into universal gate-based quantum computers—which can process logic and run general algorithms—and quantum annealers—which are used exclusively to solve optimization problems (National Academies of Sciences 2019, Egger et al. 2020). Although both technologies can theoretically solve the optimization problems explored here, currently available quantum annealers can solve much larger optimization problems than gate-based quantum computers can, and their hybrid solver counterparts can solve even larger problems (D-Wave Systems 2021).

D-Wave Systems' latest quantum annealer (also known as a quantum processing unit (QPU)), accessible via the cloud, has more than 5,000 qubits with 15-way qubit-to-qubit connectivity (the "Pegasus" topology). In addition to their quantum annealer, D-Wave Systems offers access to three hybrid solvers that make use of both classical and quantum compute resources: the CQM, the binary quadratic model, and the discrete quadratic model. Each of these solvers allows for different levels of flexibility in the construction of the models, the numbers and types of variables, and the numbers of constraints. We opted to use the CQM hybrid solver because it can handle more constraints (100,000) and binary and integer variables (500,000) than any other available quantum hardware.[5]

The CQM solver takes objectives and constraints directly as inputs; thus, the problem was submitted specifically as

$$\text{Objective}: \quad \min \sum_{\alpha} b_{\alpha} \tag{1}$$

$$\text{Subject to}: \quad b_{\alpha} + N_{\alpha}(0) + \text{mNDP}_{\alpha} + \sum_{i} \sum_{\tau \le t} f(\alpha, i)\, x_{i,\tau} \ge 0, \ \forall \alpha \tag{2}$$

$$\sum_{i} x_{i,t} = 1, \ \forall t \tag{3}$$

$$\sum_{t} x_{i,t} = 1, \ \forall i. \tag{4}$$

The variables are define in Table 1.

The economic intuition of the problem is summarized by the first constraint. An increase in the first term indicates a deterioration in the solution to the problem by requiring additional liquidity in the system above the mNDP observed up to the processing of that batch. The last term is the value of settled payments—the product of the variable that indicates if a payment has settled or not ($x_{i,t}$) and the function of the value of the payment (f). Although this term changes discretely, it is useful to think in marginal terms. A marginal increase in this term indicates that more payment value is settled, which can only be satisfied by an increase in liquidity, worsening the solution. Therefore, by searching over different sets $\{x_{i,t}\}$, the optimizer explores the trade-off between settling more payment value and increasing the liquidity necessary to do so.

4.2. From CQM Solver Results to the Optimal Queue Order

Finding a solution amounts to finding the matrix that represents the reordering of the payments from i to t

Table 1. Variables

Variable	Description
b_α	This is the participant α's liquidity allocation. When minimized, it is the amount of additional liquidity that α would need to have to permit settlement of the payments in that batch (i.e., to avoid gridlock). Equivalently, it is the increase of mNDP_α after this batch is settled.
$N_\alpha(0)$	This represents the participant α's initial balance for the batch calculated as the difference between the total payments and receipts at the start of the batch
mNDP_α	This denotes the highest net debit position a participant α has experienced from earlier batches in the day before the current batch begins
$x_{i,t}$	This is a binary decision variable that indicates whether payment i is settled ($x_{i,t} = 1$) or not ($x_{i,t} = 0$) at position t in the final queue
$f(\alpha, i)$	This is a function that returns v (the value or amount of payment i) if α is the payee, $-v$ if α is the payer, and zero otherwise

(i.e., $\{x_{i,t}\}$). For example, if there are seven payments in a queue, the CQM solver might return

$$\{x_{i,t}\} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{5}$$

This would mean that the fourth payment in the original queue should be processed first, then the payment at the top of the queue, and so on. To find these solutions, we calculate a QUBO to submit to the CQM using the objective and constraints imposed by each payment and initial $N_\alpha(0)$ and mNDP_α.

Each run of the annealer potentially ends in a different state based on the length of the anneal (analogous to "cooling time" in a physical anneal) and the underlying probability distribution of the states (the lower the energy of the final state, the higher its probability of occurring). In practice, time constraints mean that we may reach the near-optimal, but not best, solution in a given run. However, the CQM runs multiple shots because more solutions provide more confidence based on the statistics of the solution set that one of the optimal reorderings was found (there may be multiple reorderings that are equally good).

Our first step in processing the set of returned reordering solutions is to check if any violated the constraints in Equations (2)–(4) (i.e., if it is an infeasible solution). If so, those infeasible solutions are discarded. The remaining feasible solutions are then searched classically for the ones that provided the most savings, yielding our final result and the payments processing order for that batch.

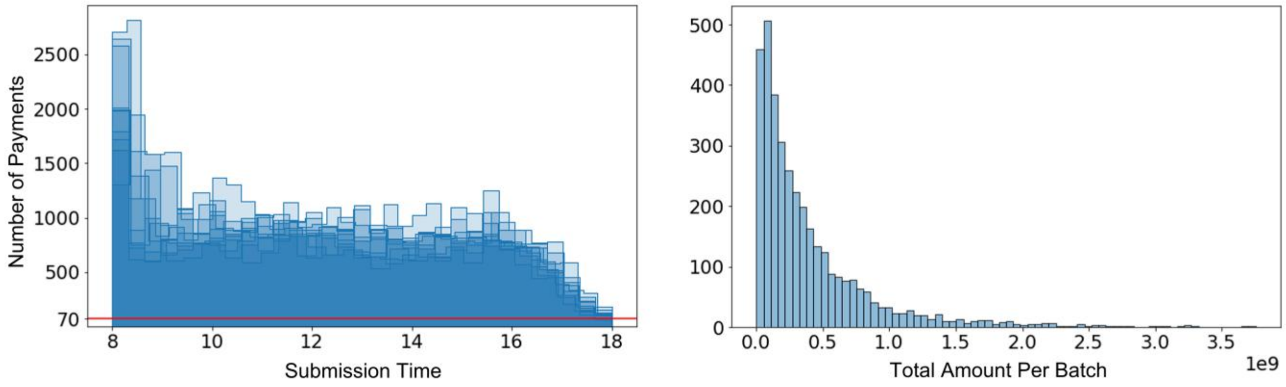
The optimality of the solution proposed by the solver is challenging to quantify for larger problems because optimal solutions become too computationally expensive to verify (the reason we turn to quantum computation in the first place). However, an approximate measure of optimality can be inferred from the number of feasible candidate solutions returned from the CQM solver. The CQM solver's default run time for batches of 70 payments, for example, provides about 50–70 solutions. As the problem gets larger with queue size, it becomes much more difficult to solve, and fewer feasible solutions come back from the CQM, despite the default run time also increasing with queue size. Thus, if the CQM solver returns a histogram with many (few) feasible results, it correlates with a better (worse) quality of its best-returned solution.

Note that using the CQM solver, each batch is processed as if it was happening in real time and only analyzed to yield aggregate results afterward. That is, we do not look ahead to use knowledge of what would come later in the day to organize each earlier batch. Thus, our method, although true to what could be achieved in real-world scenarios optimizing batch by batch, does not always result in achieving the absolute minimum liquidity requirements over the course of a full day. To achieve these would require, among other things, predicting which payments may appear in subsequent batches. As we will show, on average, institutions benefit proportionally to the value of the transactions they conduct.

5. Payments Data Sample

To test the performance of the proposed reordering algorithm, we randomly sample 30 days from Canada's HVPS settlement data between January 2015 and December 2017. From those 30 days, we exclude urgent payments from the sample and use only nonurgent payment requests because the urgent payments could have a high cost of delay. In our sample, we have approximately 23,000 transactions per business day. On some busy days, however, the Canadian HVPS processed a higher volume of transactions. Therefore, to test the benefits of a quantum optimizer on such days,

Figure 2. (Color online) (Left Panel) The Number of Payments and (Right Panel) the Total Value (in Canadian Dollars) Settled per Batch in Our Sample



we ensure that 10 high-volume days are included in our 30-day sample.

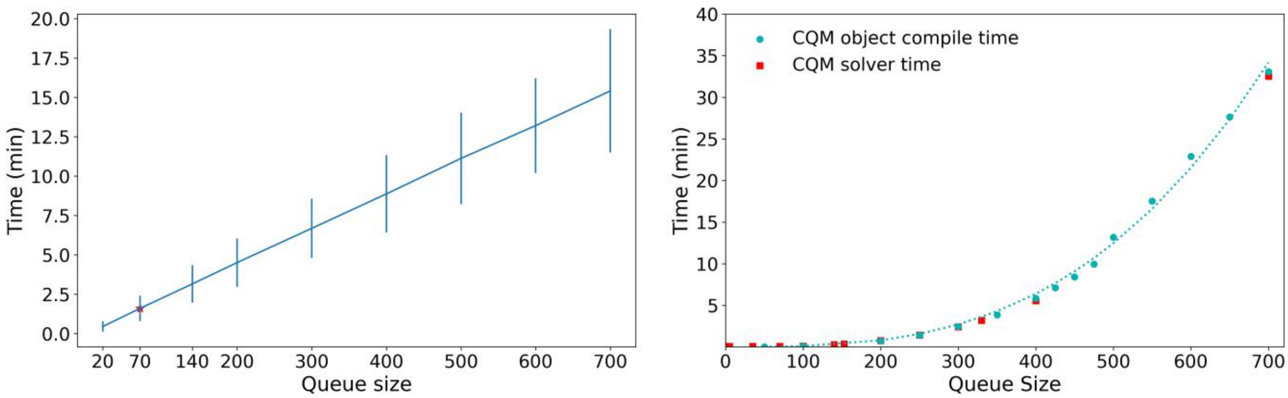
The frequency at which payment requests are submitted varies with the time of day, as shown in Figure 2, left panel. A higher volume of payments flows between 8 a.m. and 9 a.m. The number of payment requests remains steady from about 9 a.m. until about 4 p.m., before the requests taper down toward the end of the day. On the other hand, the payment values follow a Pareto distribution, maintained in the total settled values for each batch (Figure 2, right panel). The average value settled for a batch size of 70 is C$0.4 billion, but some settle as much as C$3.5 billion. The nonnormal distribution of payments makes it difficult to find bilateral or multilateral netting opportunities. Further reducing optimizability, most batches have more payees than payers.

6. Results and Discussion

This section presents the results of simulation exercises conducted on the 30-day sample of settlement data from Canada's HVPS using D-Wave Systems' CQM hybrid solver. In real-world applications, to avoid introducing added settlement delay because of backlog, the time needed to find the solution to the reordering problem should not take longer than collecting the batch of transactions.

For a queue size ranging between 20 and 700 payments, the average wait time increases linearly from 1 to 15 minutes (Figure 3, left panel). However, this wait time varies with the time of day because the frequency at which payment requests are submitted changes throughout the day. For a batch of 70 payments, the average wait time is about 90 seconds. To optimize 70 payments on CQM, the average compile and solve times are five seconds each. This time grows cubically, proportional to the number of bases of the QUBO object (Figure 3, right panel), and it limits the maximum queue size without backlog to around 140 transactions. However, there is room for potential improvement by calculating the CQM on more powerful computers with multiprocessing and reducing network latency by

Figure 3. (Color online) (Left Panel) Average Waiting Time to Fill a Queue for a Given Batch Size



Notes. The orange star highlights the batch size of 70 payments. (Right panel) Time needed to compile the CQM object (blue) and then find the solution using D-Wave Systems' hybrid CQM solver (red). The dashed blue line is the cubic fit.

calculating the problem on servers physically closer to the solvers.

6.1. Performance of Quantum Reordering Algorithm

We optimize the settlement order of payments by dividing each day into batches and solving each batch on the CQM solver. These batches are solved using each participant's mNDP and net position from the previous batch. The summary of the results for the entire sample is presented in Table 2 for a batch size of $n = 70$. Note that for this table, we run a horse race between FIFO and the CQM. They both start with the same initial conditions (for instance, the same mNDP at 8 a.m.). Consequently, every batch throughout the day is the same for both FIFO and CQM. At the end of the day (6 p.m.), we report the results (mNDP).

As summarized in Table 2, the quantum solver can provide significant end-of-day liquidity savings—averaged at C$239.9 million with a median of C$128.8 million across 30 days and following an exponential distribution. The most significant savings is C$1.26 billion (sample day 16). However, of the 9,717 batches examined across these days, 49% have no participants who moved their mNDP, and thus, no rearrangement could possibly provide liquidity savings; 25.3% of batches have an optimized rearrangement that provided liquidity savings. The remaining 25.7% have participants that grew their mNDP but are not optimizable for reasons such as the participants having only outgoing payments, FIFO already being an optimal solution, and/or a large payment that could not be offset. For greater analysis on the reordering algorithm and the influence of data on reordering algorithm performance, see Appendix B.

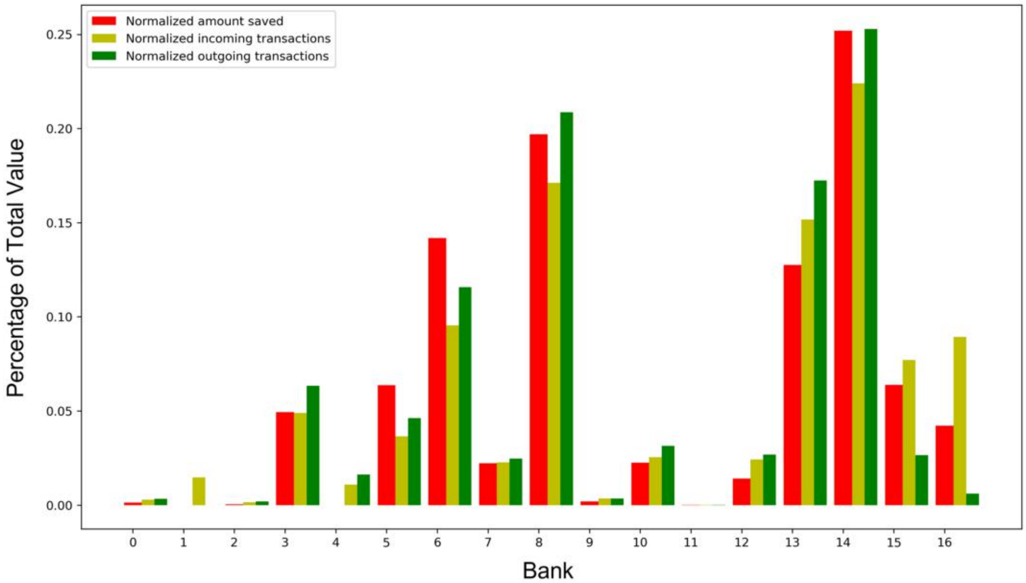
We also run the batch-by-batch horse race between CQM and FIFO, where both start with the same initial conditions for every batch to check which gives a better result. Although the full-results table is not shown, we

Table 2. Summary Statistics of the Sample Data Showing the Total Value of Payments Settled in Tranche 2 Between 8 a.m. and 6 p.m.

Sample day	Date	Value settled (C$ billion)	Total batches	Improved batches	End-of-day savings (C$ million)	Savings, %
1	2015-01-30	139.19	360	55	406.47	2.19
2	2015-02-23	101.36	240	84	542.75	3.5
3	2015-03-04	117.99	256	95	191.15	1.18
4	2015-03-23	112.44	257	102	60.53	0.37
5	2015-03-26	111.17	262	87	167.41	1.09
6	2015-03-31	163.12	387	102	30.44	0.16
7	2015-04-02	111.10	306	91	52.29	0.31
8	2015-04-15	133.56	300	85	40.38	0.26
9	2015-04-20	117.34	265	73	241.88	1.4
10	2015-05-15	128.94	317	97	24.32	0.13
11	2015-10-26	123.29	296	76	299.37	1.72
12	2015-12-16	153.42	300	85	14.40	0.07
13	2016-01-08	103.40	271	65	170.64	1.09
14	2016-01-29	150.80	386	88	997.39	4.9
15	2016-06-01	140.73	369	92	65.04	0.33
16	2016-07-15	138.24	371	78	1,264.70	6.9
17	2016-12-30	126.59	346	97	1,104.92	6.18
18	2017-02-21	120.83	311	86	206.77	1.15
19	2017-05-31	142.81	387	64	40.17	0.24
20	2017-06-01	136.88	390	70	−21.27	−0.11
21	2017-06-15	130.75	336	87	57.54	0.28
22	2017-06-30	158.00	492	89	70.76	0.35
23	2017-09-20	167.21	284	51	33.35	0.17
24	2017-09-26	101.61	276	72	213.15	1.29
25	2017-10-12	92.62	275	84	97.37	0.66
26	2017-11-15	108.92	335	95	56.13	0.26
27	2017-11-16	107.90	282	62	48.97	0.3
28	2017-11-28	91.92	290	83	371.20	2.19
29	2017-12-01	123.99	378	77	212.67	1.24
30	2017-12-22	116.70	392	80	160.92	1.12
Average		125.76	323	81	239.93	1.36

Notes. The days are sorted in descending order of total number of batches, where each batch contains $n = 70$ payments. The improved batches column shows the number in which the quantum optimizer using the CQM solver was able to reduce the aggregate mNDP compared with FIFO, and the last two columns show the total liquidity saved by the end of the day in both Canadian dollar (C$) values and percentages over FIFO.

Figure 4. (Color online) Amount of Liquidity Saved per Bank as a Percentage of Total Savings Incurred (Red) Compared with the Percentage of the Total Value of Incoming (Gold) and Outgoing (Green) Payments Using a Fixed Batch Size



find that on a typical day, an optimized batch saves C$13.25 million on average, with a median of C$0.79 million and a maximum of C$339.41 million. Overall, the maximum savings achieved by any one batch is C$1.29 billion (also on sample day 16). Occasionally, the order returned by the CQM hybrid solver results in less efficient liquidity usage than FIFO. This occurs only seven times in the 9,717 batches and never occurs more than once on any day. The loss in savings is less than C$10,000 in five of those instances and reaches as high as C$2.96 million. These instances can be attributed to stochastic noise in the quantum annealer and could easily be removed in a practical implementation of this system by immediately rerunning the affected batches (if time allowed) or defaulting to FIFO.

To examine the distribution of savings across participants, Figure 4 shows each participant's liquidity savings (using a batch size of 70) relative to its total transaction value. The percentage of the total amount saved for each participant is compared with their percentage of total incoming and outgoing transactions. The Pearson's correlation coefficients between percentage saved and percentage of incoming and outgoing transactions are $r = 0.961$ and $r = 0.967$, respectively. This high correlation implies that the savings are somewhat fairly distributed based on how much liquidity each participant moves. In other words, if a participant is involved in payments responsible for half of the total settled value in a day, they will likely see about half the savings achieved for that day.

In the upcoming sections, we will assess the performance of the CQM solver against a classical solver. We also compare the CQM reordering with some traditional LSMs and a simulation of settlement in the Canadian HVPS (called Lynx), which itself features a variety of LSMs (bypass, bilateral, and multilateral netting). The list of comparisons is outlined in Table 3.

7. Comparison with a Classical Solver

We employed the SCIP solver on an AWS t2.xlarge instance to benchmark the performance of the CQM hybrid solver against classical optimization methods, leveraging SCIP's diverse optimization techniques. We compared the reordering solutions provided by SCIP and CQM using the same reordering algorithm, the same 30-day sample, start of day balance, and mNDP. We also imposed a five-second time limit on the SCIP solver to match the solve time used by the CQM solver.

Table 3. Alternative Algorithms and Solvers for Comparing the CQM Solver Performance

Algorithm	Description
SCIP reordering	Reordering algorithm solved using the latest SCIP solver
FIFO bypass	In each batch, bypass the largest payment, and place it at the end of queue
FIFO sorting	In each batch, sort payments in ascending order of value
Lynx LSMs	LSMs of the Canadian HVPS (Lynx)

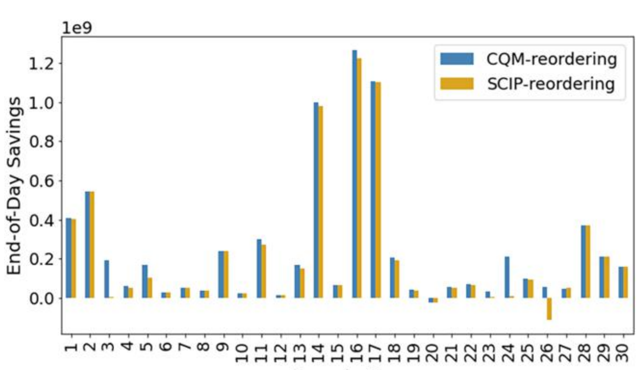
If SCIP reached its time limit without finding an optimal solution, it returned the best result achieved. In cases where no feasible solution was found, it reverted to FIFO.

The SCIP reordering solver successfully completed nearly 93% of batches within the five-second time limit. However, some batches did not finish on time, and a few remained incomplete, even with one hour of solve time. Although SCIP found solutions comparable with the CQM in most batches, it performed worse than the CQM in many cases (160 batches) and slightly better in a few (44 batches). See Appendix A for further details.

Additionally, over the 30 days in our sample, SCIP reordering achieved an average end-of-day liquidity savings of C$213.9 million compared with FIFO. This accounts for only about 87% of the savings achieved by the CQM reordering over FIFO. The distribution of end-of-day liquidity savings when comparing the two solvers is presented in Figure 5. It is worth noting that with a five-second time constraint, SCIP had slightly higher savings on just one day (about C$3 million), whereas on all other days, it exhibited decreased performance, with an average reduction in savings of about C$27 million (and a reduction exceeding C$154 in the bottom 10% of the days). These results demonstrate that the CQM can consistently provide a better solution when time is restricted while using realistic resources. Such time-bound improvements using quantum solvers have been reported earlier in the literature (Jünger et al. 2021, Tasseff et al. 2022).

In an alternate set of exercises, we allowed SCIP to run for one hour—an excess of solve time to ensure that we obtain a solution. With these solve times, one hour for SCIP and five seconds for CQM, both solvers perform similarly. The SCIP solution matches that of the CQM on most days in our sample, except that SCIP performs slightly better on nine days (with an average

Figure 5. (Color online) Comparison of End-of-Day Liquidity Savings over FIFO Using CQM Reordering and SCIP Reordering Solvers for Each Day in Our Sample Indexed by Sample Day



additional savings of C$4.51 million) but underperforms on two days (with an average decrease in savings of C$12.58 million). These results highlight the consistency of the CQM under time constraints, achieving savings similar to SCIP with a much larger solve time that would not be feasible in real-world payments system. For further details on the SCIP comparison exercises, see Appendix A.

8. Comparison with Traditional LSMs

Here, we conduct a performance comparison between the CQM reordering algorithm and conventional liquidity-saving mechanisms, including those employed in Lynx, Canada's current HVPS. For our comparative analysis, we maintained the assumption of batch settlement with a fixed delay, which means that we wait until we accumulate a batch of 70 payments and settle all payments without further delay, aligning it with the CQM setup.

The first LSM strategy is FIFO bypass—bypassing the largest payment within each batch, moving it to be last. If such a modification results in improvements over the original FIFO order for that specific batch, then the payments are settled in the suggested order. However, if no improvement is observed, the batch is settled using the original FIFO order. This process is repeated for all batches within a day in our sample, and the mNDP is computed and compared against FIFO. Using FIFO bypass, on average, across 30 days in our sample results in C$122 million in end-of-day liquidity savings over FIFO, representing about half of the savings achieved when comparing the CQM reordering against FIFO, which saves about C$240 million.

The second strategy, FIFO sorting, sorts the payments within each batch in ascending order of value and evaluates whether this new order results in improvements compared with the original FIFO order. Similar to FIFO bypass, if no improvement is observed using FIFO sorting, payments are settled following the FIFO order. On average, across 30 days in our sample, the FIFO sorting strategy yielded about C$138 million in end-of-day liquidity savings over FIFO. However, this accounts for only about 57% of the savings achieved when comparing the CQM reordering against FIFO.

The distribution of end-of-day liquidity savings against FIFO in our 30-day sample, comparing CQM reordering, FIFO bypass, and FIFO sorting is presented in Figure 6. These results suggest that the conventional ad hoc reordering strategies, like bypass and sorting, can be helpful. However, the CQM reordering obtained by searching the entire solution space, potentially including bypass and sorting, consistently yields better results, nearly doubling the savings in our sample.

Third, we compare the CQM results with the savings provided by state-of-the-art LSMs in a real HVPS. The

Figure 6. (Color online) Comparison of End-of-Day Liquidity Savings over FIFO Using CQM Reordering, FIFO Bypass, and FIFO Sorting Algorithms for Each Day in Our Sample Indexed by Sample Day

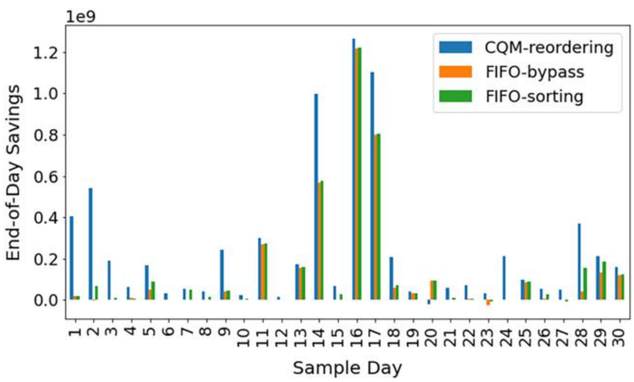
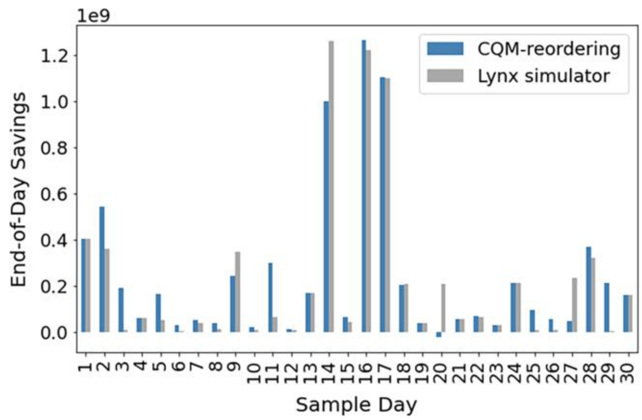


Figure 7. (Color online) Comparison of End-of-Day Liquidity Savings over FIFO Using CQM Reordering and the Simulator of the Canadian HVPS for Each Day in Our Sample Indexed by Sample Day



Canadian system, called Lynx, combines several LSMs, including bypass and bilateral and multilateral netting (see Bank of Canada 2022 for details). As in the previous exercises, we maintain the assumption of batch settlement with a fixed delay. To perform a simulation of settlement of an actual production system, we need to make an assumption of the liquidity that each participant would allocate at the beginning of the day. We assume that participants allocate as initial liquidity the CQM's mNDP (i.e., the *minimum* liquidity needed to settle all payments in the order that the CQM algorithm proposed). Subsequently, we collect a batch of 70 payments and initiate the settlement process using the simulator. If all payments are successfully settled, we proceed to the next batch. However, in the cases where not all payments are settled, we provide additional liquidity to the sending participant to address any remaining unsettled payments.

The LSMs of the Lynx system achieve on average C$224 million in end-of-day liquidity savings over FIFO, most of the time without netting in fact. This represents about 93% of the savings attained when comparing it with the CQM reordering algorithm over FIFO. The distribution of end-of-day savings across all days in our sample, comparing CQM reordering and the Lynx simulator, is presented in Figure 7. It is important to note that although there are a few days in our sample where the Lynx simulator outperforms CQM, there are also many days where CQM either performs similarly or better. These results suggest that when operating under batch settlement time constraints, the CQM reordering strategy performs better than conventional liquidity-saving mechanisms, including the ones that combine bypass and netting. However, when we remove the batch settlement time constraints and allow the Lynx simulator to delay payments for longer periods, it outperforms CQM. For further details on the Lynx comparison exercises, see Appendix C.

In summary, as shown in Figure 8, we compare the average end-of-day liquidity savings over FIFO among the five algorithms discussed earlier. This plot illustrates that under the same constraints, CQM reordering outperforms all other alternatives on average. The main takeaway from the comparison with Lynx is that exploring the entire space of possible reorderings, which had not been attempted before, compares quite favorably with a production system that combine many algorithms. This suggests adding reordering algorithms to the ones already in use in production systems, especially as quantum techniques progress and allow for larger batches. We turn to the issue scalability next.

9. Performance of Reordering Algorithm with Large Batch Size

To assess solution scalability with batch size, in this section, we conduct tests with the CQM reordering

Figure 8. (Color online) Comparison of the Average End-of-Day Liquidity Savings over FIFO in Our Sample Using the FIFO Bypass, FIFO Sorting, SCIP Reordering, Lynx Simulator, and CQM Reordering Methods

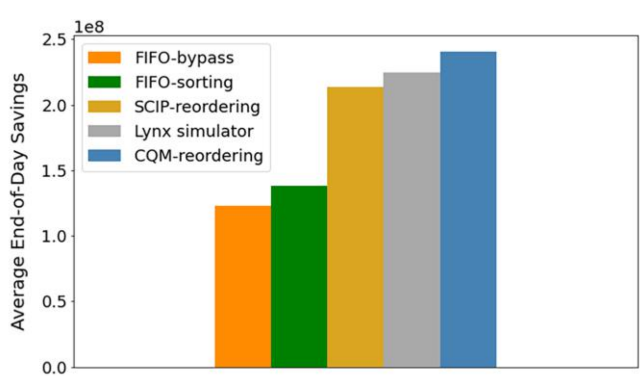


Table 4. Comparison of the Results for Two Days Using the Order Proposed by the CQM Solver and the FIFO with Batch Size $n = 140$

Date	No. of batches	Optimizable batches	No. of improved batches	Average savings (C$ million)	Median savings (C$ million)	Maximum savings (C$ million)	End-of-day savings (C$ million)
2017-09-20	142	87	50	31.46	5.93	508.41	359.56
Improvement over $n = 70$			**+13.9%**	**+9.75**	**+3.75**	**+177.19**	**+326.21**
2017-10-12	138	110	73	16.84	1.16	171.15	191.84
Improvement over $n = 70$			**+14.5%**	**+8.62**	**+0.84**	**+0.01**	**+94.47**

Note. The improvements over the same method using the batch size of 70 are shown in bold.

algorithm by doubling the batch size to $n = 140$ for two days in our sample: September 20, 2017 and October 12, 2017. As before, we run a horse race between the FIFO and CQM, where they both start with the same initial conditions and settle every batch throughout the day using FIFO and CQM orders. At the end of the day, we report the mNDP.

As detailed in Table 4, the increase in batch size produces substantial improvements in savings. Most notably, the number of batches that see higher savings over FIFO increases by around 14% for both days with respect to the number of batches that could potentially be optimized. This is because of the increased liquidity available for recycling and the increased connectivity between payers and payees in a batch's network graph.

Moreover, the increase in queue size achieves significant improvement in end-of-day savings using CQM reordering, demonstrating the scaling of the solver. Specifically, the savings compared with FIFO amounted to approximately C$360 million (a 978% increase) and C$192 million (a 97% increase). These remarkable improvements can be attributed to the superior performance of the larger queue size, the increased number of optimizable batches, and the ability to maintain savings found earlier in the day.

When we employed the SCIP solver for the same simulations with a larger batch size of 140, only about 11% of batches were successfully completed within the CQM's default solve time of 16 seconds. None of the solutions outperformed the CQM, and about 23% of batches exhibited worse performance. Furthermore, the end-of-day savings were only about C$123 million and C$36 million, less than what was achieved with a batch size of 70. This accounts for only 38% and 18% of the savings achieved by CQM on those days. However, when SCIP was allowed to run for one hour, its solution matched that of CQM. These results further underscore that the CQM solver is able to consistently outperforms SCIP when given the same solve time limits and points to more efficient problem scaling on the annealer compared with classical methods.

10. Conclusions and Future Work

Reordering algorithms are a promising avenue for improving the efficiency of financial infrastructures relying on gross settlement. We show meaningful liquidity savings in HVPS with the addition of a preprocessor, where the original payments submission sequence is reordered using a hybrid quantum solver. In a sample of 30 days of transactions from Canada's HVPS, we find improvements for queues of 70 and 140 payments, with average aggregate daily liquidity savings of C$239.93 million and C$275.70 million, respectively. These savings tend to be proportional to the participants' payments values. The quantum solution to the reordering problem proved to be more reliable, consistent, and scalable than the solution from classical computing hardware, particularly under constraints on the solve time.

Although we found much larger savings on an intraday basis (one batch reaching C$1.3 billion), the potential daily savings are limited by the timing and structure of payments between participants, especially when large outgoing payments appear in batches with few or no incoming payments to that payer. Such challenges suggest avenues for future improvements. Reordering of larger batch sizes is a clear next step. Other extensions to the reordering algorithm are to implement flexible batch sizes based on a combination of queue time, payment value, and the net balance between payers and payees. Lastly, this paper suggests the potential application of quantum in financial infrastructures that use netting, like clearing houses.

Acknowledgments

The authors thank Alex Koszegi, Tim Doyle, Peguy Pierre-Louis, and Jonathan Garven from D-Wave Systems and Maryam Haghighi and Noorain Noorani from the Bank of Canada for their help with the project. The authors also extend their gratitude to Narayan Bulusu, Rodney Garratt, Victoria Goliber, Isaiah Hull, and Hossein Sadeghi for their insightful comments. Special thanks are extended to Nellie Zhang for her help with Lynx simulations for benchmarking exercises. In addition, the authors thank discussants and participants of the following conferences for their comments and suggestions: the National Automated Clearinghouse Association

Smarter, Faster Payments Conference (2022), the Quantum FinTech Webinar Series at the University of North Carolina (2023), the Quantum Webinar Series at Innovation, Science and Economic Development (ISED) Canada (2023), the Research Webinar Series at the Bank for International Settlements (2023), and the Bank of Finland's Simulator Seminar (2023). The views expressed in this paper are solely those of the authors and do not necessarily represent the views of the Bank of Canada or other affiliations.

Appendix A. SCIP Solver Exercises

To compare the quantum solver with a classical solver, we formulate our MBO to run on SCIP solver, an open-source, versatile tool capable of solving mixed integer programming and constraint integer programming problems, among others (as cited in Anand et al. 2017, Vigerske and Gleixner 2018, and Bestuzheva et al. 2021). Its efficacy stems from a suite of methodologies. Presolving refines the initial problem by leveraging iterative reduction methods until achieving a size and complexity conducive to optimization. Domain propagation focuses on the significance of variables, whereas decomposition segments the problem for ease of processing. The cornerstone of SCIP, however, is its branch-and-bound strategy, wherein the problem is systematically dissected into smaller components based on varying variable values. This strategy incorporates pseudocosts and primal heuristics to inform and refine the decision-making process.

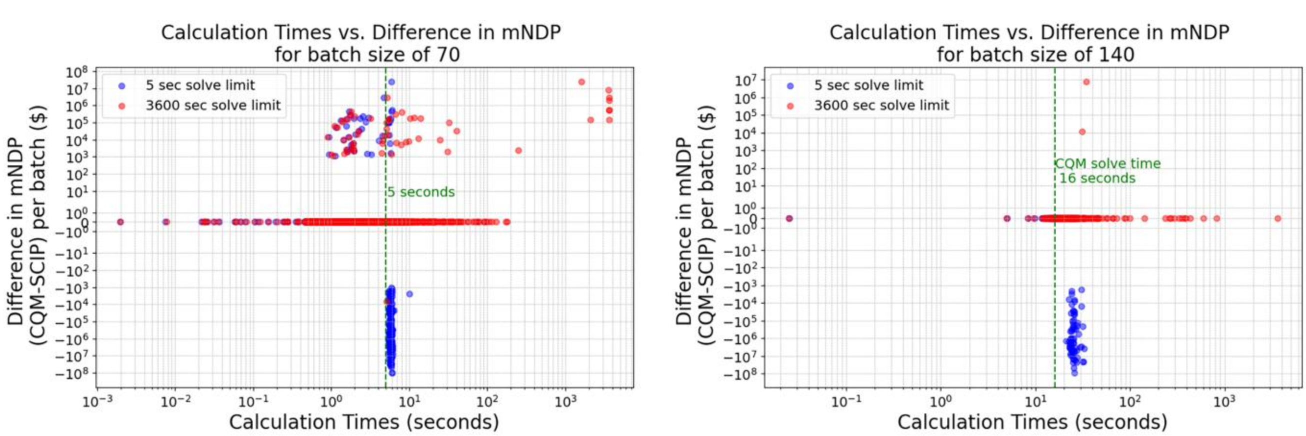
The intricate mechanism of SCIP's branch-and-bound strategy deserves special mention. Pseudocosts commence with default values and are then continuously refined based on observed changes in the objective function. Strong branching, albeit computationally intensive, offers more precision by determining the exact objective function impact. SCIP's decision on whether to lean on the pseudocosts or engage in strong branching hinges on the reliability criteria, notably the frequency of a variable's branching. This ensures that the chosen variable for branching is the one with the most potential influence on the objective function.

Supplementing these core methodologies, SCIP also incorporates advanced techniques, like conflict analysis and cutting planes, to fine-tune its optimization approach. Conflict analysis delves deep into the reasons for infeasible solutions, shedding light on potential issues and ensuring a more streamlined search. On the other hand, cutting planes narrows the solution space by integrating additional linear inequalities, which not only refine the solution parameters but also hasten the overall optimization process.

SCIP is used to explore a classical solver's solve time and solution quality and compare it with quantum annealing. As mentioned in the text, the calculation time is divided into presolve and solve time. To compare with the CQM, a limit was placed on the solve time. We used the same solve times as CQM (5 and 16 seconds for batches of 70 and 140, respectively) as well as one hour—an excess of solve time to derive a solution. If the solve time was reached with no optimal solution, the solver was forced to return the best it had achieved, and if it had not found a feasible solution, it returned FIFO. The same 9,717 optimization problems solved by CQM were presented to SCIP (the same set of transactions, initial balance, and starting mNDP).

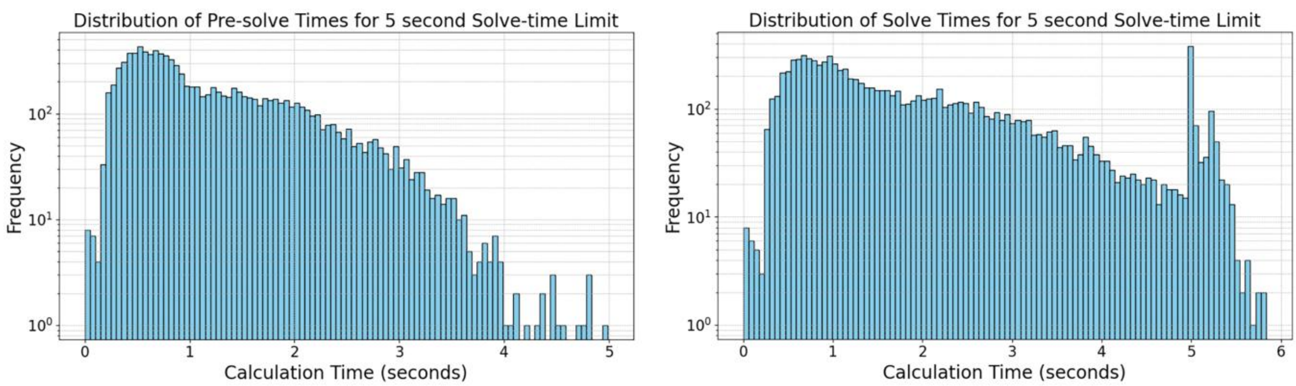
For a batch size of 70 and a time limit of an hour, 93% finished within the first 5 seconds (calculation time mean: 6.89 seconds, median: 2.06 seconds). However, five batches did not finish within the hour, and four had a presolve time of over 5 seconds (average presolve time: 1.09 seconds, median: 0.8 seconds). SCIP found a better solution than CQM in 58 batches (red circles in Figure A.1, left panel), exemplifying that quantum annealing is not an exact solver but given enough time, tends to the ground-state optimal solution. One batch returned a worse solution when SCIP did not find a feasible solution, so it returned FIFO. For a batch size of 70 and a time limit of 5 seconds, the average calculation time was 3.17 seconds (average presolve time: 1.2 seconds, solve time: 1.9 seconds) (Figure A.2). Forty-four batches did better and 160 batches did worse than CQM (blue circles in Figure A.1, left panel). Of the batches that did worse, all reached the five-second limit and were forced to conclude. When the SCIP

Figure A.1. (Color online) Representation of Solution Quality of SCIP vs. CQM



Notes. The difference in mNDP (CQM – SCIP) is plotted vs. calculation time. The blue and red dots are five-second and one-hour SCIP solve times, respectively, and the green lines indicate CQM solve time. (Left panel) Batch size 70: 9,717 batches. (Right panel) Batch size 140: 280 batches.

Figure A.2. (Color online) The Distribution of (Left Panel) Presolve Time and (Right Panel) Solve Time (in Seconds)



solution was worse, the average difference was \$4,637,877 compared with \$725,466 when SCIP was better.

When the batch was doubled to 140 and the solve time was restricted to 16 seconds, the same as the CQM solve time, no solution did better and 66 (23%) did worse using SCIP (with an average decrease in performance of \$5.90 million) (blue circles in Figure A.1, right panel). When given an hour, SCIP concluded in all but 1 of 280 batches within the time. However, 88.9% of batches needed over 16 seconds of solve time (average: 52.4 seconds, median: 21.7 seconds). CQM found the same mNDP as SCIP in all but two batches.

The comparative performance in end-of-day savings through cumulative batch optimization between CQM, SCIP with an hour limit, and SCIP constrained to five seconds is illustrated in Figure 5. An overarching observation reveals that CQM and SCIP granted an hour perform similarly. However, not one solver significantly dominates the other. Given an hour, SCIP outperforms the CQM on nine days (with an average additional savings of \$4.51 million) but underperforms on two days (with an average decrease in savings of \$12.58 million), matching it on all the other days. With a time constraint of five seconds, SCIP had slightly higher savings on one day (\$2.99 million); however, on all other days, it had diminished performance (an average decrease in savings of \$27.80 million). The representation of these data suggests that no solver can consistently guarantee optimal savings. The five-second SCIP may suffice for CQM only in specific scenarios but is unreliable for engendering meaningful savings. Although a one-hour SCIP duration matches or surpasses, there remain instances when CQM outperforms. A closer examination of the results conveys an important message that the quality of batch-by-batch optimization influences end-of-day savings. Consequently, although there are sporadic instances where CQM overshadows SCIP, it is crucial to interpret these findings with a discerning lens given the variegated conditions and constraints that each solver operates under.

Appendix B. Data Influence on Algorithm Performance

To help visualize the queue reordering process in a typical batch of 70 payments (on September 20, 2017), each participant's debit and credit balance evolution is shown in Figure B.1. The chart in the upper panel of Figure B.1 uses the FIFO order, and the chart in the lower panel of Figure B.1 is for the order proposed by quantum optimization. Each line represents each bank's balance as the transactions are completed sequentially in the prescribed order. Because we optimize the aggregate mNDP, we want to limit the depth to which the participants drop below zero. In this example, the mNDP for Bank 14 was originally \$17 million. The optimization rearranges the transactions to prevent Bank 14 (the blue lines in Figure B.1) from taking any debit position while not penalizing other participants. Doing so and reordering other banks' payments saves the system \$18 million in aggregate mNDP over these 70 transactions. However, it should be noted that not all transactions in this batch can be optimized. Two examples are Bank 13 (the teal lines in Figure B.1) and Bank 16 (the orange lines in Figure B.1), which have only outgoing payments; therefore, no scope exists for optimizing their mNDP through queue reordering any of their transactions in this batch.

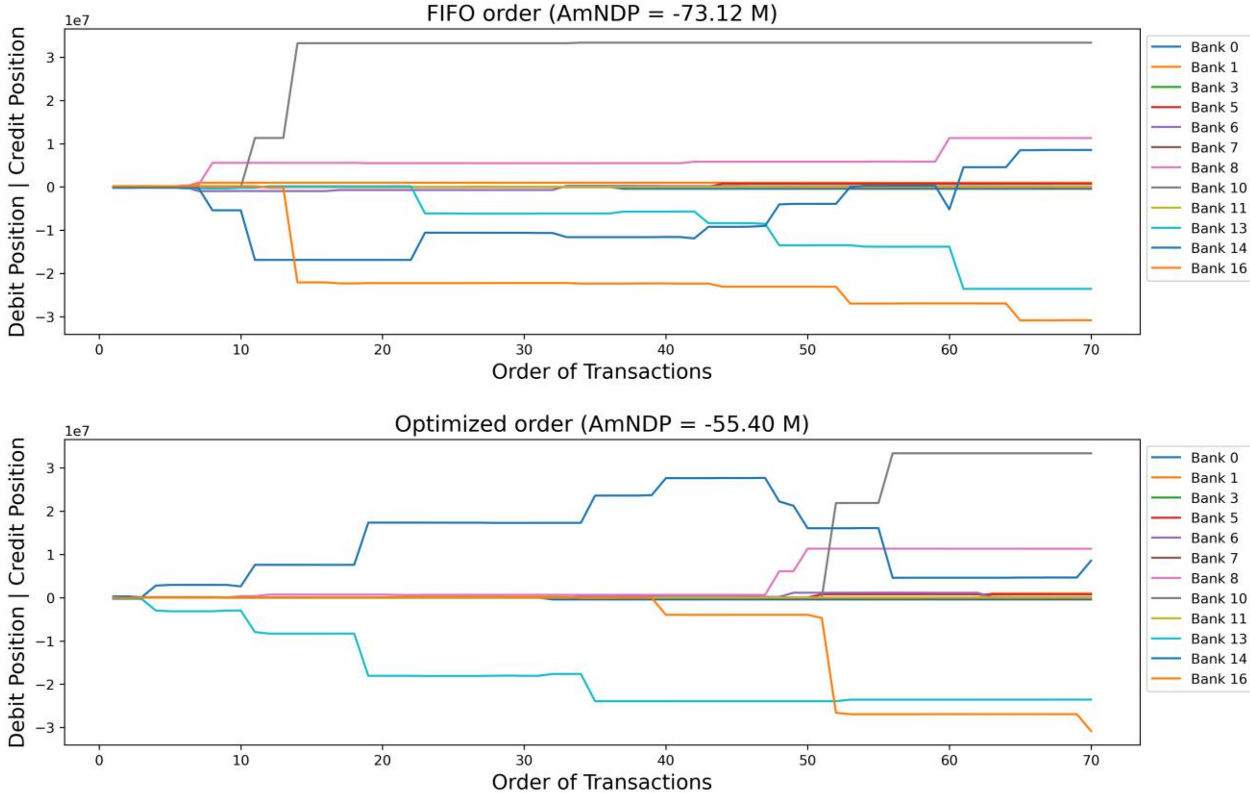
As an example, we show the evolution of the aggregate mNDP for a typical day (September 20, 2017) in our sample in Figure B.2 for a batch size of $n = 70$. At the start of the day, as participants send their payments, we see a sharp rise in aggregated mNDP. As the day progresses, however, we see a slower increase in aggregated mNDP. For many batches during the day, no participant had moved its mNDP, and thus, no rearrangement could provide liquidity savings. These are seen as flat sections in the aggregate mNDP plot.

Sometimes, a large payment (or group of payments) in a single batch can cause the aggregate mNDP to converge for the CQM solver and FIFO, erasing savings that the CQM had achieved earlier in the day. Examples of this can be seen at 14:20 and more extremely, at 16:05 in Figure B.2. This erasure cannot be avoided without increasing the queue size or using a more complicated queue-building algorithm that allows payments to move between queues. Clearly, for a chosen batch size, there is significant heterogeneity across the batches, and there are limitations on the improvements that can be achieved with this simple preprocessing setup.

Appendix C. Lynx Simulator Exercises

Lynx is Canada's HVPS. It is an RTGS system with central queuing and LSMs intended to reduce liquidity usage. Lynx has various mechanisms, such as bypass and netting

Figure B.1. (Color online) The Change in Participating Banks' Credit or Debit Position for a Given Queue of 70 Transactions Immediately After Each Payment Settlement for September 20, 2017

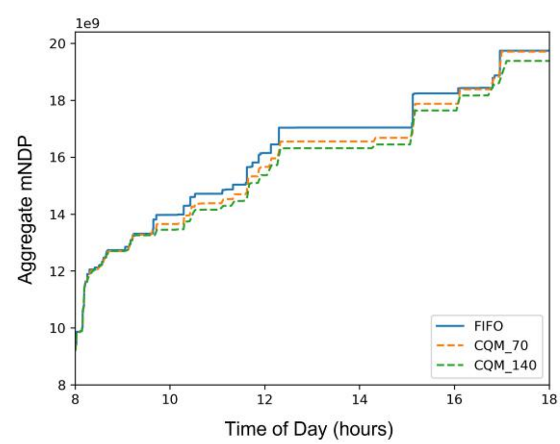


Note. The upper panel shows the FIFO order ($73 million Aggregated max net debit position (AmNDP)), and the lower panel shows the order proposed by the quantum CQM solver ($55 million aggregate mNDP).

algorithms, that allow participants to optimize liquidity usage. At the core of its LSM is a payment-offsetting algorithm called *Gridlock Buster* that runs intermittently and allows for concurrent settlement of payments in the queue at each run. As a result, when payments are netted against each other on a multilateral level, the resulting multilateral net debit positions of all sending participants involved in the batch are smaller than FIFO, where payments are settled in the original order on a gross basis; hence, less liquidity is needed to fund these debit positions. Nonetheless, in Lynx, the majority of payments sent to the LSM are settled immediately upon submission (Rivadeneyra and Zhang 2022, Desai et al. 2023). Using the Lynx simulator developed in Rivadeneyra and Zhang (2022), we conduct a performance comparison between the CQM reordering algorithm and the traditional LSMs offered by Lynx. To conduct this comparative analysis, in the first set of simulations, we modify the simulator to align with the CQM setup. This adaptation maintains the assumption of batch settlement with a fixed delay, where we wait until we accumulate a batch of 70 payments and then settle all payments without introducing additional delays. Following this, in the second set of exercises, we relax the batch settlement assumption, allowing payments to be delayed and settled continuously.

To initiate the simulation exercises on a given day in our sample, we need to allocate an initial liquidity amount for each participant. We start by utilizing the CQM's mNDP for that day as the reference point. Subsequently, we collect a batch of 70 payments and initiate the settlement process using the simulator. We opt for these initial levels of liquidity because they represent the minimum amount required to

Figure B.2. (Color online) The Aggregate Max Net Debit Position vs. Time of the Day for September 20, 2017



Note. The blue line shows the original payment order (FIFO); the orange and green lines, respectively, show the orders returned from the quantum annealers, with the day divided into batch sizes of 70 and 140 payments.

settle all payments in the order proposed by the CQM reordering algorithm. Therefore, employing the same initial liquidity levels as required by the CQM reordering algorithm in conjunction with an identical settlement delay allows for meaningful comparisons.

In the initial set of exercises utilizing the adapted simulator, we follow this procedure; if all payments in a given batch are successfully settled, we proceed to the next batch. However, in cases where not all payments are settled, we provide additional liquidity to the sending participant to facilitate the settlement of their remaining unsettled payments. We repeat this until the end of the day. In this setup, averaged across 30 days in our sample, we achieve approximately C$224 million in end-of-day liquidity savings over FIFO. This amount represents approximately 93% of the savings achieved when comparing it with the CQM algorithm over FIFO. These results suggest that when operating under batch settlement time constraints, CQM reordering performed better compared with Lynx.

In the second set of exercises, we remove the restrictions related to batch settlement, allowing the simulator to settle payments continuously. In this scenario, no intraday liquidity is provided if payments are delayed within a batch; instead, the simulator can potentially delay these payments indefinitely. When tested on the same 30-day sample, on average, 46 payments experience delays, with an average delay time of just 55 seconds. The maximum delay observed is 15 minutes, and there are no rejected payments at the end of the day. These statistics indicate that the Lynx simulator, operating without time constraints and using the same level of liquidity as CQM, outperforms the CQM reordering algorithm with time restrictions. The primary advantage arises from continuous settlement, enabling some payments to be delayed for longer time and settled in subsequent batches.

Subsequently, in another set of exercises, we reduce the initial liquidity for each participant at the start of the day by some fraction *alpha*. Starting from the minimum level of liquidity required by the CQM reordering algorithm (mNDP of CQM), if we reduce liquidity by 10% using *alpha* = 0.9, the simulation results demonstrate that the average number of payments delayed each day in our sample increased to 1,300 payments, and the maximum daily delay averaged around three hours. Additionally, on average, about five payments were rejected each day. This indicates that by reducing liquidity by 10% from the CQM's minimum required level, we run the risk of rejecting a few payments and long delays for many payments. Next, we test the Lynx simulator by reducing initial liquidity by 2.5% (i.e., using *alpha* = 0.975). Even with this small reduction, we still observed an average of two payments being rejected, with some payments experiencing delays exceeding an hour. These results suggest that the CQM solution (i.e., mNDP computed using CQM reordering) is very close to the minimum needed level in Lynx to balance liquidity optimization without risking payment rejections.

Endnotes

[1] To get a sense of their importance, Canada's HVPS processes payment values equivalent to annual Gross Domestic Product every week. In 2021, Canadian financial institutions participating in HVPS allocated on average Canadian (C) $15 billion in daily liquidity to process around 40,000 payments each day.

[2] For examples of these algorithms, see the description of the HVPS of the United Kingdom and Canada (Bank of England 2021, Bank of Canada 2022).

[3] To see this, take a system with two participants, A and B, having to pay each other one dollar twice. If the order of payments is AABB (each letter representing a dollar payment to the other), then the minimum liquidity needed is two dollars because A needs to allocate two dollars of initial liquidity before B can pay back. If the order is ABAB, the liquidity needed is only one dollar. Thanks to a referee for suggesting this example.

[4] An alternative type of payment system is a deferred net settlement. In these systems, payments are submitted by participants and accepted by the system, but they are not settled in that moment. Instead, payment exposures are accumulated and settled at given intervals after calculating the net positions of the participants, therefore creating credit risk. See Norman (2010) for an overview of LSMs.

[5] The main downside to using CQM hybrid solvers is that the details of their inner workings are not precisely controlled. On D-Wave Systems' QPU, parameters, such as annealing time, number of reads, and annealing path, can be controlled.

References

Alexandrova-Kabadjova B, Badev A, Bastos SB, Benos E, Cepeda-Lopez F, Garratt R, Heijmans R, et al. (2023) Intraday liquidity around the world. BIS Working Paper No. 1089, Bank of International Settlements, Basel, Switzerland.

Anand R, Aggarwal D, Kumar V (2017) A comparative analysis of optimization solvers. *J. Statist. Management Systems* 20(4):623–635.

Atalay E, Martin A, McAndrews J (2010) Quantifying the benefits of a liquidity-saving mechanism. FRB of New York Staff Report No. 447, Federal Reserve Bank of New York, New York.

Bank of Canada (2022) An overview of Lynx, Canada's high-value payment system. Technical report, Bank of Canada, Ottawa, ON.

Bank of England (2021) Liquidity saving mechanism user guide. Technical report, Bank of England, London.

Bech M, Garratt R (2003) The intraday liquidity management game. *J. Econom. Theory* 109(2):198–219.

Bestuzheva K, Chmiela A, Müller B, Serrano F, Vigerske S, Wegscheider F (2023) Global optimization of mixed-integer nonlinear programs with SCIP 8. *J. Global Optim.*, ePub ahead of print December 14, https://doi.org/10.1007/s10898-023-01345-1.

Bestuzheva K, Besançon M, Chen W-K, Chmiela A, Donkiewicz T, van Doornmalen J, Eifler L, et al. (2021) The SCIP Optimization Suite 8.0. ZIB-Report No. 21-41, Zuse Institute Berlin, Berlin.

Braine L, Egger D, Glick J, Woerner S (2021) Quantum algorithms for mixed binary optimization applied to transaction settlement. *IEEE Trans. Quantum Engrg.* 2:1–8.

Castro PS, Desai A, Du H, Garratt R, Rivadeneyra F (2020) Estimating policy functions in payment systems using reinforcement learning. Staff Working paper, Bank of Canada, Ottawa, ON.

D-Wave Systems (2021) Hybrid solver for constrained quadratic models. D-Wave Systems White Paper. Accessed February 1, 2021, https://www.dwavesys.com/media/rldh2ghw/14-1055a-a_hybrid_solver_for_constrained_quadratic_models.pdf.

Davey N, Gray D (2014) How has the liquidity saving mechanism reduced banks' intraday liquidity costs in CHAPS? Bank of England Quarterly Bulletin Q2, Bank of England, London.

Desai A, Lu Z, Rodrigo H, Sharples J, Tian P, Zhang N (2023) From LVTS to Lynx: Quantitative assessment of payment system transition in Canada. *J. Payments Strategy Systems* 17(3):291–314.

Diehl M, Schollmeyer U (2009) Liquidity-saving mechanisms: Quantifying the benefits in TARGET2. Hellqvist M, Laine T, eds. *Diagnostics for the Financial Markets—Computational Studies of Payment System* (Bank of Finland, Helsinki), 411–430.

Egger DJ, Gambella C, Marecek J, McFaddin S, Mevissen M, Raymond R, Simonetto A, Woerner S, Yndurain E (2020) Quantum computing for finance: State-of-the-art and future prospects. *IEEE Trans. Quantum Engrg.* 1:1–24.

Farhi E, Goldstone J, Gutmann S, Lapan J, Lundgren A, Preda D (2001) A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem. *Science* 292(5516):472–475.

Fernández-Villaverde J, Hull IJ (2023) Dynamic programming on a quantum annealer: Solving the RBC model. NBER working paper, National Bureau of Economic Research, Cambridge, MA.

Finnila A, Gomez M, Sebenik C, Stenson C, Doll J (1994) Quantum annealing: A new method for minimizing multidimensional functions. *Chemical Phys. Lett.* 219(5):343–348.

Galbiati M, Soramaki K (2010) Liquidity-saving mechanisms and bank behaviour. Bank of England working paper, Bank of England, London.

Garratt RJ (2022) An application of Shapley value cost allocation to liquidity savings mechanisms. *J. Money Credit Banking* 54(6):1875–1888.

Hidary JD (2019) *Quantum Computing: An Applied Approach* (Springer, Cham, Switzerland).

Hull I, Sattath O, Diamanti E, Wendin G (2020) Quantum technology for economists. Preprint, submitted December 9, http://dx.doi.org/10.2139/ssrn.3745608.

Jünger M, Reinelt G, Rinaldi G (1995) The traveling salesman problem. Ball MO, Magnanti TL, Monma CL, Nemhauser GL, eds. *Handbooks in Operations Research and Management Science*, vol. 7 (Elsevier, Amsterdam), 225–330.

Jünger M, Lobe E, Mutzel P, Reinelt G, Rendl F, Rinaldi G, Stollenwerk T (2021) Quantum annealing vs. digital computing: An experimental comparison. *ACM J. Experiment. Algorithmics* 26:1–30.

Jurgilas M, Martin A (2013) Liquidity-saving mechanisms in collateral-based RTGS payment systems. *Ann. Finance* 9:29–60.

Kadowaki T, Nishimori H (1998) Quantum annealing in the transverse Ising model. *Phys. Rev. E* 58(5):5355–5363.

Martin A, McAndrews J (2008a) An economic analysis of liquidity-saving mechanisms. Working paper, Federal Reserve Bank of New York, New York.

Martin A, McAndrews J (2008b) Liquidity-saving mechanisms. *J. Monetary Econom.* 55(3):554–567.

Mittelmann H (2023) Decision tree for optimization software. Technical report, Arizona State University, Tucson.

National Academies of Sciences (2019) *Quantum Computing: Progress and Prospects* (The National Academies Press, Washington, DC).

Norman B (2010) Liquidity saving in real-time gross settlement systems: An overview. *J. Payments Strategy Systems* 4(3):261–276.

Parekh O, Wendt J, Shulenburger L, Landahl A, Moussa J, Aidun J (2016) Benchmarking adiabatic quantum optimization for complex network analysis. Working paper, Bank of International Settlements, Basel, Switzerland.

Rehfeldt D, Koch T, Shinano Y (2022) Faster exact solution of sparse Maxcut and Qubo problems. *Math. Programming Comput.* 15: 445–470.

Rivadeneyra F, Zhang N (2020) Liquidity usage and payment delay estimates of the new Canadian high-value payments system. Technical Report No. 9, Bank of Canada, Ottawa, ON.

Rivadeneyra F, Zhang N (2022) Payment coordination and liquidity efficiency in the new Canadian wholesale payments system. Bank of Canada Staff Discussion Paper No. 2022-3, Bank of Canada, Ottawa, ON.

Şeker O, Tanoumand N, Bodur M (2022) Digital annealer for quadratic unconstrained binary optimization: A comparative performance analysis. *Appl. Soft Comput.* 127:109367.

Skavysh V, Priazhkina S, Guala D, Bromley TR (2023) Quantum Monte Carlo for economics: Stress testing and macroeconomic deep learning. *J. Econom. Dynamics Control* 153:104680.

Stamatopoulos N, Egger DJ, Sun Y, Zoufal C, Iten R, Shen N, Woerner S (2020) Option pricing using quantum computers. *Quantum* 4:291.

Steane A (1998) Quantum computing. *Rep. Progress Phys.* 61(2):117.

Tasseff B, Albash T, Morrell Z, Vuffray M, Lokhov A, Misra S, Coffrin C (2022) On the emerging potential of quantum annealing hardware for combinatorial optimization. Working paper, Bank of International Settlements, Basel, Switzerland.

Vigerske S, Gleixner A (2018) Scip: Global optimization of mixed-integer nonlinear programs in a branch-and-cut framework. *Optim. Methods Software* 33(3):563–593.

Woerner S, Egger DJ (2019) Quantum risk analysis. *npj Quantum Inform.* 5:15.