IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 31 August 2023, accepted 10 September 2023, date of publication 18 September 2023,
date of current version 28 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3316727

RESEARCH ARTICLE

Pairs-Trading System Using Quantum-Inspired Combinatorial Optimization Accelerator for Optimal Path Search in Market Graphs

KOSUKE TATSUMURA, (Member, IEEE), RYO HIDAKA, JUN NAKAYAMA, TOMOYA KASHIMATA, AND MASAYA YAMASAKI
Corporate Research and Development Center, Toshiba Corporation, Kawasaki 212-8582, Japan

Corresponding author: Kosuke Tatsumura (kosuke.tatsumura@toshiba.co.jp)

ABSTRACT Pairs-trading is a trading strategy that involves matching a long position with a short position in two stocks aiming at market-neutral profits. While a typical pairs-trading system monitors the prices of two statistically correlated stocks for detecting a temporary divergence, monitoring and analyzing the prices of more stocks would potentially lead to finding more trading opportunities. Here we report a stock pairs-trading system that finds trading opportunities for any two stocks in an N-stock universe using a combinatorial optimization accelerator based on a quantum-inspired algorithm called simulated bifurcation. The trading opportunities are detected through solving an optimal path search problem in an N-node directed graph with edge weights corresponding to the products of instantaneous price differences and statistical correlation factors between two stocks. The accelerator is one of Ising machines and operates consecutively to find multiple opportunities in a market situation with avoiding duplicate detections by a tabu search technique. It has been demonstrated in the Tokyo Stock Exchange that the FPGA (field-programmable gate array)-based trading system has a sufficiently low latency (33 μs for N=15 or 210 pairs) to execute the pairs-trading strategy based on optimal path search in market graphs.

INDEX TERMS Pairs-trading, trading system, real-time system, custom circuit, FPGA, combinatorial optimization, tabu search, Ising machine, simulated bifurcation, quantum-inspired.

I. INTRODUCTION

A financial market with high efficiency and high liquidity is where investors can execute high-volume trading at fair values, at any time without significantly impacting the market prices. The concept of arbitrage is defined in [1] as the simultaneous purchase and sale of the same, or essentially similar, security in two different markets for advantageously different prices. Arbitrage opportunities can arise as a result of demand shocks and arbitragers bring temporarily deviated prices (hereafter, mispricing) to fundamental (fair) values. Arbitrage enforces the law of one price and thereby improves the efficiency of financial markets [2]. Recent studies [3], [4] have also shown that arbitrage provides liquidity.

Pairs-trading strategy is categorized as a statistical arbitrage and profits from temporary mispricing of statistically correlated stocks [5]. The strategy monitors the performance of two historically correlated stocks for detecting the moment when one stock relatively moves up while the other relatively moves down (possibly temporarily), and at that moment simultaneously takes a short (selling) position of the outperforming stock and a long (buying) position of the underperforming one with each position having the almost same amount of transaction, betting that the spread between the two would eventually converge. The strategy is market-neutral, i.e., adaptable to various market conditions (uptrend, downtrend, or sideways) by keeping the net exposure low.

Various variants of pairs-trading that differ in how to identify comoving stocks and how to decide the timing of position opening have been proposed and summarized

The associate editor coordinating the review of this manuscript and approving it for publication was Fabian Khateb.

in [6], involving distance approach, cointegration approach, time-series approach, stochastic control approach and other approaches (including machine learning approaches like recent one using long short-term memory networks [7]). Those, not necessarily mutually exclusive, can contribute to improving the market efficiency and liquidity by detecting the different trading opportunities (occurrences of mispricing).

To analyze the collective structure of a stock market, market graphs have been proposed and utilized [8], [9], [10], where the nodes correspond to the stocks and each edge (or edge weight) between two nodes represents the relationship of the two stocks defined based on correlation factors [8], [9] or more generalized risk-measures [10]. Moreover, higher-order networks that directly includes triadic or polyadic interactions [11], [12] have been studied for representing the markets [13], [14], [15]. Graph/network analysis methods such as partitioning, clustering, coloring, and path search may give insights into the collective structures/behaviors of the stocks. Many of those methods are formulated as combinatorial (or discrete) optimization problems and belong to the nondeterministic polynomial time (NP)-hard class in computational complexity theory [16].

Ising machines are hardware devices that solve the ground (energy minimum)-state search problems of Ising spin models and can be of use for quickly obtaining the optimal (exact) or near-optimal solutions of NP-hard combinatorial optimization problems [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34]. The Ising problem belongs to the NP-hard class [16], [35]; a variety of notoriously hard problems including many graph analysis methods can be represented in the form of the Ising problem [16]. Various hardware implementations of Ising machines based on quantum mechanics, optics, or analog/digital electronics have been studied extensively in recent years [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34].

The Ising machine can be applied to automated trading systems [36], [37], [38], [39], [40] including ones executing pairs-trading and may enable the detection of trading opportunities based on the computationally-hard analysis of market graphs within the lifetime of the opportunities (the duration time until the market situation changes). Automated trading systems become increasingly important in financial markets [41], [42] and the trading strategy enabled with emerging computing methodologies would complement the functionality of the market or contribute to mitigating the herding behaviors in financial markets [43]. High-speed trading strategies based on combinatorial/discrete optimization and trading systems utilizing Ising machines as in [44] have been, however, not extensively studied. Furthermore, the execution capability of such a trading system in terms of response latency needs to be validated in the actual market since the duration time of the trading opportunity of a strategy is determined by the activities of other trading entities.

Here we propose a pairs-trading strategy based on an optimal path analysis in market graphs and show through real-time trading that the strategy is executable with an automated pairs-trading system using an embedded Ising machine for the optimal path search.

The market graph for N tradable stocks (an N-stock universe) is an N-node fully-connected directed graph with edge weights corresponding to the products of instantaneous price differences and statistical correlation factors between two stocks. The trading opportunities (temporary mispricing of statistically correlated pairs) are detected by an optimal path analysis (a sort of collective evaluation) of the N-node market graph. As the embeddable Ising machine, we use a combinatorial optimization accelerator based on a quantum-inspired algorithm called simulated bifurcation (SB) [17], [18], [19], [20], [21]. The algorithm of SB, derived in 2019 [17] through classicizing a quantum-mechanical Hamiltonian describing a quantum adiabatic optimization method [45] and improved in 2021 [19], is highly parallelizable and thus can be accelerated with parallel processors such as FPGAs (field-programmable gate arrays) [18]. FPGA-based SB machines (SBMs) are suitable for high-speed trading systems because they can be integrated in an FPGA together with other system components to shorten the system-wide latency. The embedded SBM used in this work is customized for the proposed strategy and operates consecutively to find multiple trading opportunities in an instantaneous market situation with avoiding duplicate detections by a tabu search technique. To examine the execution capability of the system, we compare the real-time transaction records of the system in the Tokyo Stock Exchange (TSE) with a backcast simulation of the strategy assuming the orders issued are necessarily filled.

The rest of the paper is organized as follows. In Sec. II (trading strategy), we describe the proposed strategy and formulate the optimal path search in the form of quadratic unconstrained binary optimization (QUBO) mathematically equivalent to the Ising problem. Sec. III (system) describes the architecture of the system and its implementation details. Sec. IV (experiment) describes the transaction records in the TSE and the execution capability of the system. Sec. V discusses the extensibility of the proposed strategy and system in relation with market representation as higher-order networks. Sec. VI concludes the paper.

II. TRADING STRATEGY

A. PATH SEARCH-BASED PAIRS-TRADING

The proposed strategy determines open pairs (a pair of long and short positions in two stocks to be taken) by an optimal path analysis of an N-node market graph representing a relative relationship in the prices of N stocks. The evaluation of a pair is based on not only the direct path but also any bypass paths. Multiple pairs can be chosen in an instantaneous market situation.

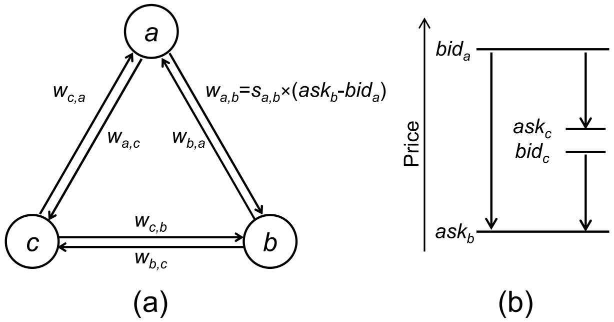


FIGURE 1. (a) Market graph for an N-stock universe ($N = 3$). (b) A relationship of *bid* and *ask* values regarding with the direct path ($a \rightarrow b$) and bypass path ($a \rightarrow c \rightarrow b$) evaluations of pair (a, b).

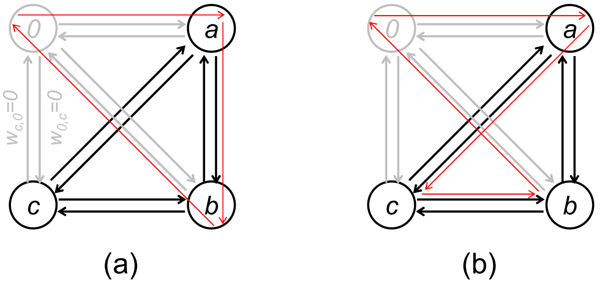


FIGURE 2. (a) Market graph with the dummy node ($i = 0$) for an N-stock universe ($N = 3$). (a) A cyclic path ($0 \rightarrow a \rightarrow b \rightarrow 0$), represented by red arrows, corresponding to the direct path for the pair (a, b). (b) A cyclic path ($0 \rightarrow a \rightarrow c \rightarrow b \rightarrow 0$) corresponding to the bypass path for the pair (a, b).

The market graph for an N-stock universe (Fig. 1a) is a directed graph in which an edge (i, j) corresponds to a trading pair that takes a short position of ith stock and a long position of jth stock and is distinguished from the edge (j, i). The weight $w_{i,j}$ of an edge (i, j) is defined by

$$w_{i,j} = s_{i,j} \times (ask_j - bid_i) \tag{1}$$

where $s_{i,j}$, ask_j, and bid_i are, respectively, the similarity factor between ith and jth stocks, the best ask for jth stock, and the best bid for ith stock. *ask* and *bid* are normalized by the base price on the day. $s_{i,j}$ is based on the average value for the last five business days of the dynamic time warping (DTW) distance [46] of the price sequences (per day) of ith and jth stocks and is normalized to be in [0, 1]. When the buying price of a long position (ask_j) is relatively lower than the selling price of a short position (bid_i) in the two stocks with a large similarity ($s_{i,j}$), $w_{i,j}$ is negative and its absolute value is large.

In the market graph, two nodes connected by the minimum-weight one-way directed path are considered to correspond to the best trading opportunity. A pair of nodes can be selected based on a bypass path rather than the direct path. In the case of Fig. 1, the pair (a, b) is evaluated for both the direct path ($a \rightarrow b$) and the bypass path ($a \rightarrow c \rightarrow b$). The bypass path corresponds to concurrently taking the pair (a, c) and pair (c, b) positions, leaving the pair (a, b) position as a result of the cancellation of buying and selling the stock c (the direct and bypass paths correspond to the same open pair). If not considering the similarity factors, the sum of $w_{a,c}$ and $w_{c,b}$ (bypass) is always higher than $w_{a,b}$ (direct) by the bid-ask spread of the stock c (transit nodes on the bypass) (see Fig. 1b). However, considering the similarity factors, the sum of $w_{a,c}$ and $w_{c,b}$ can be lower than $w_{a,b}$. In this case, the evaluation of pair (a, b) is represented by the sum of the weights on the bypass path. This bypass evaluation (or collective evaluation) partially complements the incompleteness of the representation of similarity coming from characterizing time series data as a scalar value and prevents us from missing the trading opportunity. The evaluation value (weight sum) of a pair selected by the optimal path analysis is compared with a threshold for determining the opening of the pair.

The number of lots per order for a stock (L_i) is determined to make the amount of transaction (A_{trans}) common for all tradable stocks by rounding with considering the minimum tradable shares per order (a lot) of the stock (S_i^{min}) and the base price on the day (p_i^b); $L_i = round(A_{\text{trans}}/S_i^{\text{min}} p_i^b)$. The number of intraday positions is controlled to be within a maximum number (P_{max}) and all positions are closed (unwind) before the close of the day. Duplicate pair positions are not allowed. When the pair (a, b) has been ordered (opened), another order of the same pair (a, b) has been forbidden, but other pairs including (a, c) and (c, b) are orderable and the edge (a, b) is passable for bypass evaluation.

Consider a subgroup of stocks (for an example, a, b, and c) that are correlated one another. If the price of one in the subgroup (assume a in the example) deviates largely (drops in the example) while the prices of the remaining ones do not deviate, multiple pairs related to the deviating one [pairs (b, a) and (c, a) in the example] are highly evaluated at the moment and, as well as the best pair [pair (b, a) in the example], the second-best pair [pair (c, a) in the example] can be worth betting (can have an evaluation value beyond the threshold). To our backcast simulation (see Sec. IV), a temporary price deviation of one stock in a subgroup involving correlated stocks gives good trading opportunities. For finding multiple opportunities in a market situation, the optimal path analysis is repeated. We need a sort of tabu search technique to avoid repeatedly finding the solution that has been found.

B. FORMULATION

The problem to find a pair of two nodes connected by the minimum-weight directed path (direct or bypass) from any two nodes in the N-node market graph is formulated in the form of the QUBO. A tabu search technique using a tabu list ($T_{i,j}$) is implemented in the formulation.

After adding a dummy node ($i = 0$) with edge weights of zero ($w_{k,0} = w_{0,k} = 0, {}^{\forall}k > 0$) in the market graph (Fig. 2), we seek a cyclic (directed) path giving the minimum

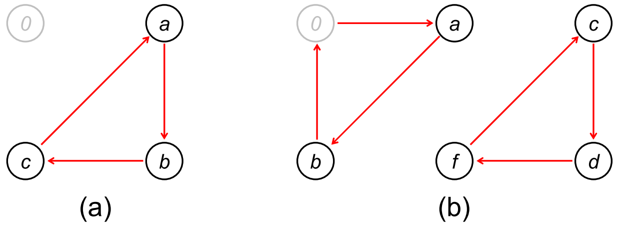


FIGURE 3. Solutions to be excluded by the verification. (a) A cycle without the dummy node. (b) Split cycles.

weight. Let the node next (/previous) to the dummy node in the cyclic path correspond to the short (/long) positions of a pair trade. As shown in Fig. 2, a pair (a, b) is represented by both the cyclic path $(0 \to a \to b \to 0)$ and the cyclic path $(0 \to a \to c \to b \to 0)$ with the different weight sums. The former (/latter) representation corresponds to the direct (/bypass) evaluation of the pair (a, b). Clockwise and anticlockwise cycles (ex. $0 \to a \to b \to 0$ and $0 \to b \to a \to 0$) are distinguished.

Define a decision (binary) variable $b_{i,j}$ as taking value 1 if the corresponding edge (i, j) is in the chosen cycle and 0 otherwise. The cost function to be minimized is defined by

$$H_{\text{cost}} = \sum_{i,j} w_{i,j} b_{i,j}. \tag{2}$$

The constraints for cyclic directed paths and the tabu search are represented as a penalty function expressed by

$$\begin{aligned} H_{\text{penalty}} = & \sum_{i} \sum_{j \neq j'} b_{i,j} b_{i,j'} + \sum_{j} \sum_{i \neq i'} b_{i,j} b_{i',j} \\ & + \sum_{i} \Big(\sum_{j} b_{i,j} - \sum_{j} b_{j,i} \Big)^2 \\ & + \sum_{i,j} b_{i,j} b_{j,i} + \sum_{i,j} T_{i,j} b_{0,j} b_{i,0}. \end{aligned} \tag{3}$$

The first (/second) term forces the outflow (/inflow) of each node to be 1 or less. The third term forces the inflows and outflows of each node to be equal. The fourth term forbids traversing the same edge twice in different directions. The fifth term forbids choosing the pairs in the tabu list $T_{i,j}$. Constraint violations increase the penalty, with $H_{\text{penalty}} = 0$ if there are no violations. Note that an entry $T_{i,j}$ in the tabu list induces a penalty for the state $(b_{0,j} = b_{i,0} = 1)$ but not for the states $(b_{0,j} = 1$ and $b_{i,0} = 0)$, $(b_{0,j} = 0$ and $b_{i,0} = 1)$, and $(b_{i,j} = 1)$.

The total cost function (H_{QUBO}) is a linear combination of H_{cost} and H_{penalty},

$$H_{\text{QUBO}} = \sum_{i,j,k,l} Q_{i,j,k,l} b_{i,j} b_{k,l} = m_c H_{\text{cost}} + m_p H_{\text{penalty}}, \tag{4}$$

where m_c and m_p are positive coefficients. The Ising machine searches for the bit configuration $\{b_{i,j}\}$ that minimizes the quadratic cost function H_{QUBO}.

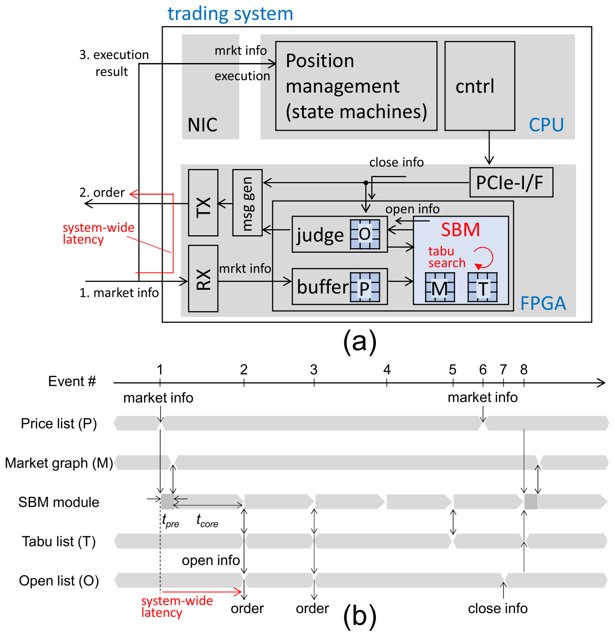


FIGURE 4. System architecture (a hybrid FPGA/CPU system). (a) Block diagram. (b) Timing chart.

The tabu search technique was introduced to enhance the search efficiency upon the multiple executions of the Ising machine for finding multiple opportunities in a market situation under the constraint of forbidding duplicate positions. The procedure and timing of registering and deregistering entries in the tabu list are described in Section III. In the QUBO formulation, the number of decision variables for an N-stock universe is $N(N + 1)$ and the size of the solution space (all possible points of the decision variables) is $2^{N(N+1)}$, including constraint violation solutions. We use a heuristic method (an Ising machine) to solve the QUBO problems. Hence, the verification of solutions is necessary and implemented in the system as a function other than the Ising machine. In addition, the penalty function, Eq. (3), gives no penalty to the two cases (a cycle without the dummy node and split cycles) shown in Fig. 3. Those solutions are excluded by the verification. Note that those solutions are not advantageous in the evaluation of the cost function, Eq. (2).

III. SYSTEM

To accelerate the decision of opening positions and the issuance of orders after receiving a market feed (informing the change of *ask* or *bid* of a stock), the submodules related to the position opening are, in an FPGA, hardwired (instantiated as custom circuits) and inlined as a task pipeline from a receiver (RX) to a transmitter (TX), which are functional without the intervention of a software processor (CPU). The SBM module involved in the pipeline is an inline-type accelerator (not a look-aside type one), featuring a consecutive execution operation and a tabu search function. The management of the positions including the decision of closing positions is carried out by the CPU (software processing). Overall, the system is a hybrid FPGA/CPU system.

A. ARCHITECTURE

Figure 4 (a) shows the block diagram of the hybrid FPGA/CPU system. The system components in the FPGA part are, in the order of data flow, a receiver (RX), a price buffer (P) that accommodates the price list of *ask* and *bid* for the N tradable stocks, the SBM module, a judgment module with a memory unit for the open list (O), a message generator, and a transmitter (TX). The SBM module includes two memory units for a market graph (M) and a tabu list (T), a preprocessing unit (*pre*) for preparing the market graph, and a core processing unit (*core*) for the discrete optimization. Those components are implemented as independent (not synchronized) circuit modules, which are connected by directed streaming data channels with FIFO (first-in-first-out) buffers. The CPU part controls the whole system and manages the positions using state machines for opened positions (see APPENDIX A). The market information (including the changes in *ask* or *bid*) is received by both the FPGA and CPU parts. The order (buying/selling) packets are issued only from the FPGA part. The execution-result packets informing the results (fill/lapse) of the orders are received by the CPU part. The FPGA and CPU parts are connected with the peripheral component interconnect-express (PCIe) bus.

Figure 4 (b) shows the timing chart for the operation of the SBM module when representative events (Events 1 to 8) happen. When no event happens for a certain time, the SBM module is idling (polling to the FIFO buffers from the price buffer and judgment modules). When a market feed arrives (Event 1), the SBM module immediately starts the preprocessing. The preprocessing unit receives the $2N$ data of *ask* and *bid* and then generates the $N(N-1)$ data of weight $w_{i,j}$ (market graph, M) with referring to a memory unit for similarity $s_{i,j}$ which is updated once a day before the trading hours. Afterward, the SBM module starts the main (core) processing (the optimal path analysis). Then the SBM module verifies the solution (the path found) in terms of the constraint violations (including the cases of Fig. 3) and compares the evaluation of the path found with the threshold. If the verification and evaluation pass, the SBM module registers the pair in the tabu list T and concurrently informs it as an open candidate to the judgment module (Event 2). The judgment module determines the open positions by finally checking them in terms of $P_{\max}$ (the maximum number of intraday positions) and other control signals, then registers them in the open list O and issues order packets via the message generator (Event 2).

Here, the judgment module registers the open pair position in the open list O when the opening is decided (before the issuance of orders) and deregisters them when the closing of the pair position is confirmed with the message from the CPU part. When the number of pair positions is decreased, the judgment module informs the updated open list O to the SBM module, which forces the SBM module to refresh the tabu list T by copying the open list O for avoiding duplicate positions.

At the timing of Event 2, the SBM module starts the main processing again (the consecutive execution operation) without refreshing the tabu list (already up-to-date) and preprocessing (no new market feed arrives), resulting in another order at the timing of Event 3 (the SBM module could find another tradable path efficiently due to the tabu list). When the SBM does not output an effective solution (Event 4), the tabu list T and the open list O are not updated. Note that considering the pair based on a direct path (/ a bypass path) corresponding to the ineffective solution may satisfy the threshold if it is evaluated on a bypass path (/ a direct path), we designed that in this case (Event 4) the pair is not registered in the tabu list. When the SBM outputs an effective solution but it is rejected by the judgment module [for example, due to excess positions ($> P_{max}$)] (Event 5), the tabu list T is updated but the open list O is not updated. When a new market feed (Event 6) (or a close confirmation information, Event 7) arrives, the market graph M (or the tabu list T) is updated by the preprocessor (or by copying the open list), at the beginning of the next execution of the SBM module (Event 8).

As seen in Event 5, the SBM module determines registering in the tabu list without considering the decision by the judgment module. This design contributes to reducing the latency (not to incorporate the feedback latency from the judgment module). Note that the registration in the tabu list in the case of Event 5 seems to be undesirable (might miss a trading opportunity) but the over-registration in the tabu list does not matter practically because the tabu list is updated when the positions decrease (Event 8).

B. CUSTOMIZED SBM CORE CIRCUIT

The core processing unit (*core*) is architecturally similar to the basic SBM circuit design [18] but partially modified for the specific QUBO problem described in Sec. II-B. The weight $w_{i,j}$ in Eq. (2) and tabu list $T_{i,j}$ in Eq. (3) are stored in separate memory units (the M memory and T memory in Fig. 4), which are directly accessed by the SBM computation units. Based on the specific pattern of the coupling matrix Q, inefficient parts (the products with zero) in the pairwise interaction computation in the SB algorithm are omitted.

In the consecutive execution operation, the SBM module repeats the main processing (simulating the time-evolution of a coupled oscillator network) with different initial states generated by an internal random number generator (RNG), Xorshift RNG [47]. This contributes to efficiently finding another good solution even when the market graph M and the tabu list T are not updated (Event 4). The latency of the RNG is hidden by overlapping the operations of the SBM core and the RNG; the RNG generates an initial state for the next execution of the SBM core while the SBM core is processing.

C. IMPLEMENTATION

We implemented the system described in Sec. III-A with a CPU server with a network interface card (NIC) and an FPGA

	SBM module
Architecture	
algorithm version	ballistic SB (bSB)
machine size (# of spins)	256
computational precision	32-bit floating point
Resource (core+pre.)	
ALM [count]	52,241 + 46,919
BRAM [count]	450 + 145
BRAM [bits]	262,016 + 1,095,936
DSP [count]	340 + 170
Implementation	
F_{sys} [MHz]	233
Cycles per SB step [cycle]	138
Cycles for pre. [cycle]	216
Performance (N_{step} = 50)	
$t_{pre+core}$ [µs]	30.6
t_{core} [µs]	29.6
t_{pre} [µs]	0.9

(a)

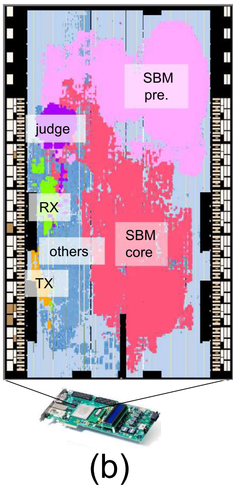


FIGURE 5. **System implementation. (a) Architecture and implementation details of the SBM. (b) Placement of functional modules in the FPGA.**

board having another network interface (see APPENDIX B for details).

Figure 5 (a) shows the architecture and implementation results of the SBM module for 15-stock universes [N=15 stocks, $N(N-1)$=210 pairs]. The numbers of nodes and edges (directed) in the market graphs supported are, respectively, 16 and 240, including the dummy node explained in Sec. II-B. Among three variants of simulated bifurcation (adiabatic, ballistic, and discrete SBs) [19], ballistic SB is adopted in this work, with the SB parameters of N_{step}=50 and dt=0.65. The machine size (the number of spins) is 256 spins with a specific spin-spin connectively for the QUBO problem described in Sec. II-B, and the computation precision is 32-bit floating point. Figure 5 (b) shows the result of the placement of system modules in the FPGA. The SBM module (*core* and *pre*) is dominant, and the circuit resources used are listed in Fig. 5 (a). The system clock frequency determined as a result of the circuit synthesis, placement, and routing is 233 MHz. The clock cycles of the SB main processing (*core*) and preprocessing (*pre*) are 6,900 cycles per run (138 cycles per SB step) and 216 cycles per run, respectively. The computation time (the module latency) per run ($t_{pre} + t_{core}$) is 30.6 μs, where the SBM core processing is dominant (t_{core}=29.6 μs). The system-wide latency from the market feed arrival to the order packet issuance depicted in Fig. 4(b) as a red arrow is 33 μs (including the latencies of the RX, price buffer, SBM, judgment, message generator, and TX modules).

The speed performance and solution accuracy of FPGA-based ballistic SB have been evaluated using various NP-hard benchmark problems and demonstrated to be competitive with other state-of-the-art Ising machines in [19]. Here we show the capability of the SBM module to find the trading opportunities of the proposed strategy within the time constrains using the historical market data of the TSE. Assuming an N=15 universe (210 pairs) selected in the bank/insurance sections as in Sec. IV, we sampled 619 market situations that include at least one trading opportunity each in the period from Aug. 1, 2017 to Jul. 31, 2020 and, for each market situation, examined the number of the trading opportunities and the duration time (the time until one of *ask*/*bid* prices in the N=15 universe changes). Also, we examined how many trading opportunities the consecutive-execution SBM finds in the duration time out of all the possible trading opportunities, with varying the time per SBM execution. SB simulates the bifurcation process of coupled oscillators with the predetermined number of discrete time steps (N_{step}). The solution accuracy improves with increasing N_{step} [17], [19] but the time per SBM execution also increases (the number of SBM executions in the duration of an instantaneous market situation decreases). Figure 6 shows the ratio of the number of trading opportunities found by the SBM module to the number of all the possible opportunities in the 619 samples as a function of N_{step}. The ratio, or the probability of finding the trading opportunities within the time constraints, is maximized at N_{step}=50 to be 96.0 %.

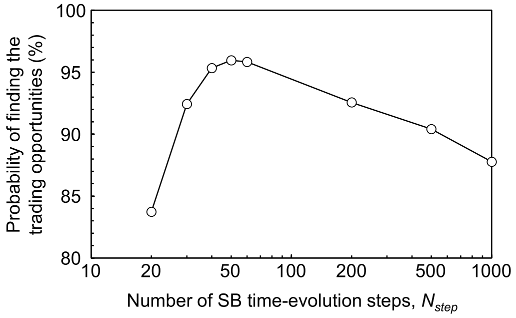


FIGURE 6. **Probability of finding the trading opportunities within the time constraints versus the number of SB time-evolution steps (N_{step}). As N_{step} (or the time per SBM execution) increases, the probability of finding better solutions (satisfying constrain conditions) per SBM execution increases but the number of SBM executions in the duration of an instantaneous market situation decreases.**

IV. EXPERIMENT

The trading system described in Sec. III was installed at the JPX Co-location area of the TSE and operated through real-time trading to examine whether the strategy based on the consecutive optimal path searches in the N-node market graph in Sec. II is executable. The trading results are compared with a backcast simulation of the strategy assuming the orders issued are necessarily filled.

The proposed strategy determines the opening of positions based on an instantaneous market situation (a price list of *ask* and *bid* for the N-stock universe). Because of the latency of a system that executes the strategy and the activities of other trading entities, the orders issued are not necessarily filled at the *ask*/*bid* prices used for the decision-making. We developed a simulator that processes the historical market feeds provided by the TSE and emulates the internal state

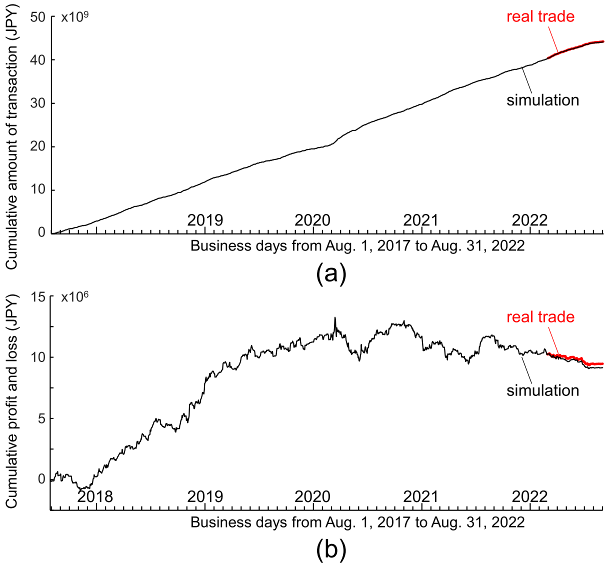


FIGURE 7. Performance of the strategy. (a) Cumulative amount of transaction in billion JPY and (b) cumulative profit and loss in million JPY. Simulation data is from Aug. 1, 2017, to Aug. 31, 2022 (1,239 business days). Real trade data is from Mar. 1, 2022, to Aug. 31, 2022 (125 business days), adjusted with the simulation at the first day.

of the trading system. The simulator assumes that the orders issued are necessarily filled at the intended prices.

Figures 7 (a) and (b) show the cumulative values of the amounts of transactions per day and the profit and loss (including *ask-bid* spread costs and commission) per day for real-time trading (red line) and backcast simulation (black line) with fixed strategic parameters of N=15 (210 pairs), $P_{\max}$=16, and A_{trans}=1.5 million Japanese yen (JPY). The 15 stocks were selected from the bank/insurance sections in terms of high liquidity. The simulation data is from Aug. 1, 2017, to Aug. 31, 2022. The real trade data is from Mar. 1, 2022, to Aug. 31, 2022, being adjusted with the simulation at the first day.

The annualized return and risk over the simulation period (approximately 5 years) are, respectively, 7.5 % and 9.5 % for an investment of 24 million JPY ($A_{\text{trans}} \times P_{max}$). The Sharpe ratio of the strategy is 0.79, where the Sharpe ratio [48] is, in this work, the ratio of the mean to the standard deviation of the return (the profit and loss per period for an investment) from a strategy as in [49]. The strategy proposed can be profitable (a positive annualized return) for the long term (approx. 5 years), especially has shown a high annualized return of 18.5 % for the period of Aug. 1, 2017, to Feb. 28, 2020, before the COVID-19 pandemic.

The cumulative value of the amount of transaction by the system (3,817,201,458 JPY) over the experiment (750 hours of real-time trading) is coincident well (+2.6 %) with the simulation value (3,719,389,258 JPY). The fill rate at the intended prices was 93.4 % and the remaining included the fills at less-favorable prices and the lapses. Most of the lapses occurred just after the opening of the morning sessions. In this experiment, when the order for one of the paired stocks lapses, the position for the other (if the order is filled) is also closed immediately for experimental simplicity (see APPENDIX A), allowing the system to execute more transactions under the constrain of the maximum number of positions ($P_{\max}$). This is the reason for the increased transaction amount observed in the experiment. Based on the good agreement in the cumulative transaction amounts and detailed comparison analysis of transactions between the experiment and simulation, we conclude that the strategy proposed is executable with the trading system with a latency of 33 μs.

Figure 8 (a) and (b) show typical transaction behaviors by the trading system observed on Mar. 10, 2022, and Apr. 1, 2022, respectively. The number of the market feeds informing the changes of *ask/bid* of stocks in the $N(= 15)$-stock universe on Mar. 10 (/Apr. 1) were 1,101,741 (/1,007,773), which arrived at intervals of 18.0 ms (/19.6 ms) on average.

On Mar. 10, 2022, the system decided the opening of the pair position (8750, 8355) [selling code 8750, buying code 8355] at 9:12:14 AM in JST (734 seconds after 9:00:00 AM) based on the evaluation of the bypass path (8750 → 8303 → 8355) found by the SBM module. It was confirmed by the backcast simulation that the evaluation value of the direct path (8750 → 8355) did not satisfy the threshold, meaning that this trading opportunity was missed if the bypass path was not evaluated for decision-making. On that day, both the prices of codes 8750 and 8355 were moving up (uptrend), but the relative difference of the prices (the spread) of the pair position decreased after the position opening, resulting in the profitable closing of the pair position before the end of the trading hours [Fig. 8 (a)].

On Apr. 1, 2022, the system decided the opening of the pair positions (8304, 8355) [selling code 8304, buying code 8355] and (8308, 8355) [selling code 8308, buying code 8355] at 9:12:11 AM in JST (731 seconds after 9:00:00 AM) based on the evaluation of the direct paths (8304 → 8355) and (8308 → 8355). The two pair positions were found by the consecutive execution operation of the SBM module in the instantaneous market situation (before the market situation changed). On that day, the prices of codes 8308, 8304, and 8355 were, overall, moving up (uptrend), but the spreads of the pair positions decreased after the position opening, resulting in the profitable closing of the pair positions before the end of the trading hours [Fig. 8 (b)].

V. DISCUSSION

The market graph used in this work is a directed/weighted graph with edges describing pairwise (dyadic) relationships between stocks. The collective evaluation of the stocks for detecting the trading opportunities is incorporated through the network analysis (path search) including the bypass evaluation and the consecutive search with the tabu search technique.

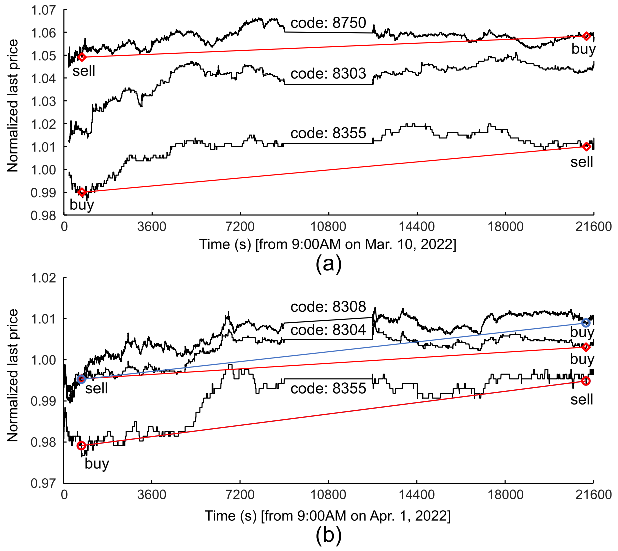


FIGURE 8. **Typical transaction behaviors of the trading system on (a) Mar 10, 2022, and (b) Apr. 1, 2022. (a) The open decision (8750, 8355) was made based on the evaluation of the bypass path (8750→ 8303→ 8355). (b) Multiple pairs (8308, 8355) and (8304, 8355) were opened in a market situation.**

Higher-order networks [11], [12] directly represent triadic or polyadic relationships in groups of three or more nodes and go beyond the paradigm of graph modeling dyadic relationships. Representing financial markets as higher-order networks has been studied for analyzing and predicting the structure and dynamics of the markets [13], [14], [15]. For example, the dependency between two stocks under the condition of the third stock can be represented by the triadic terms [13], [14], [15].

Simulated bifurcation (SB) numerically simulates the time evolution of a nonlinear oscillator network according to the Hamilton equations of motion. An SB time-evolution step updates the position and momentum of each oscillator based on the joint force caused by the interactions with the other oscillators (the joint force is the gradient of the potential/cost function). The cost function that SB evaluates is not limited to the second-order function (dyadic interactions) and may include the higher-order terms (triadic and polyadic interactions). Kanao et al. have extended SB to higher-order cost functions and showed that the higher-order SB outperforms the second-order SB with additional oscillators (higher-order cost functions can be transformed to second-order ones by adding auxiliary variables) when solving Boolean satisfiability problems including third-order cost functions [21].

Combining market representation as higher-order networks with SB-based network analysis would be one of the interesting future works.

VI. CONCLUSION

We proposed a pairs-trading strategy that finds trading opportunities for any two stocks in an N-stock universe through solving an optimal path search problem in market graphs and have demonstrated with the real-time transaction records in the TSE that the strategy is executable in terms of response latency with the automated trading system using the SB-based embeddable Ising machine for the market graph analysis.

The market graph for the N-stock universe is an N-node fully-connected directed graph with each edge weight corresponding to the product of instantaneous price difference and dynamic time warping (DTW) distance-based similarity between a pair of stocks. In the graph, two nodes connected by the minimum-weight one-way directed path selected from among all possible direct and bypass paths (a collective evaluation of the graph) are considered to correspond to the best trading opportunity. The optimal path search is formulated in the form of QUBO and consecutively executed by the SB-based Ising machine to find multiple trading opportunities in an instantaneous market situation with avoiding duplicate detections by a tabu search technique.

The automated trading system is a hybrid FPGA/CPU system. The FPGA part (hardware processing) decides the opening of a pair of long/short positions using the SB-based Ising machine customized to the specific QUBO and then issues the corresponding orders, while the CPU part (software processing) manages the opened positions (including the decision of closing positions). The system-wide latency from the market feed arrival to the order packet issuance is 33 μs for N=15 or 210 pairs.

The trading system was installed at the JPX Co-location area of the TSE and operated for a real-time trading period of 125 business days or 750 hours. The real-time transaction records were compared with a backcast simulation of the strategy assuming the orders issued are necessarily filled at the intended prices. Based on the good agreement in the cumulative transaction amounts and detailed comparison analysis of transactions between the experiment and simulation, we have concluded that the response latency of the system with the SB-based Ising machine is sufficiently low to execute the pairs-trading strategy based on optimal path search in market graphs.

Automated trading systems with embedded Ising machines (simulated bifurcation machines) would be applicable to the strategies based on various network analyses of market graphs or higher-order networks defined by various return/risk measures and other trading strategies that rely on high-speed discrete optimization.

APPENDIX A
POSITION MANAGEMENT

The position management module manages $N(N-1)$ state machines corresponding to all the pairs. Fig. 9 shows the states and transitions of the state machine. Initially the pair position (i, j) has been closed (*closed* state). When an execution packet (informing that the order of one of the stock pair is filled) is received, the state shifts to *opening* state ($T1$ transition) and then stays waiting for the remaining

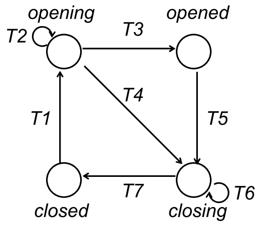


FIGURE 9. State diagram for a state machine for the pair position management.

results to be received ($T2$). If the fill of the orders for the pair is confirmed as intended, the state shifts to *opened* ($T3$). Otherwise (unintended), the state shifts to *closing* ($T4$). The management module always monitors the prices (*bid* and *ask*) of all the tradable stocks and detects the convergence of the spread when opened (the confirmation of a profit more than a threshold) for the opened pair. If the closing condition is satisfied, the state shifts to *closing* ($T5$). In the *closing* state, the state stays waiting for the related positions to be all closed ($T6$); the management module issues the orders for closing via the message generator in the FPGA and then (if necessary) repeats ordering until all the positions are closed. If the closing of the positions is confirmed, the state shifts back to *closed* ($T7$).

APPENDIX B
IMPLEMENTATION DETAILS

An FPGA board and a high-speed network interface card (NIC) are mounted on a host server with dual CPUs (Intel Xeon Silver 4215R) and DDR-DRAM modules (384 GB). The FPGA (Intel Arria 10 GX 1150 FPGA) on the board has 427,200 adaptive logic modules (ALMs) including 854,400 adaptive look-up-tables (ALUTs, 5-input LUT equivalent) and 1,708,800 flip-flop registers, 2,713 20Kbit-size RAM blocks (BRAMs), and 1,518 digital signal processor blocks (DSPs). The system components in the FPGA described in Section III were coded in a high-level synthesis (HLS) language (Intel FPGA SDK for OpenCL, ver. 18.1). The FPGA interfaces including a PCIe IP (PCIe Gen3 × 8), a 10 Gbps Ethernet PHY IP and communication IPs (RX, TX) were written in Verilog HDL and incorporated in the board support package (BSP).

ACKNOWLEDGMENT

The experiment in the Tokyo Stock Exchange was conducted under a joint project between Toshiba Corporation and Dharma Capital. K.K. The authors would like to thank Ryosuke Iio and Kohei Shimane for fruitful discussions and technical support.

CONFLICTS OF INTEREST

K.T., R.H., and M.Y. are included in inventors on two U.S. patent applications related to this work filed by the Toshiba Corporation (no. 17/249353, filed 20 February 2020; no. 17/565206, filed 29 December 2021). The authors declare that they have no other competing interests.

REFERENCES

[1] W. F. Sharpe, G. J. Alexander, and J. V. Bailey, *Investments*, 4th ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 1990.

[2] A. Shleifer and R. W. Vishny, "The limits of arbitrage," *J. Finance*, vol. 52, no. 1, pp. 35–55, Mar. 1997, doi: 10.1111/j.1540-6261.1997.tb03807.x.

[3] D. Gromb and D. Vayanos, "Limits of arbitrage," *Annu. Rev. Financial Econ.*, vol. 2, no. 1, pp. 251–275, Dec. 2010, doi: 10.1146/annurev-financial-073009-104107.

[4] D. Rösch, "The impact of arbitrage on market liquidity," *J. Financial Econ.*, vol. 142, no. 1, pp. 195–213, Oct. 2021, doi: 10.1016/j.jfineco.2021.04.034.

[5] E. Gatev, W. N. Goetzmann, and K. G. Rouwenhorst, "Pairs trading: Performance of a relative-value arbitrage rule," *Rev. Financial Stud.*, vol. 19, no. 3, pp. 797–827, 2006, doi: 10.1093/rfs/hhj020.

[6] C. Krauss, "Statistical arbitrage pairs trading strategies: Review and outlook," *J. Econ. Surveys*, vol. 31, no. 2, pp. 513–545, Apr. 2017, doi: 10.1111/joes.12153.

[7] A. Flori and D. Regoli, "Revealing pairs-trading opportunities with long short-term memory networks," *Eur. J. Oper. Res.*, vol. 295, no. 2, pp. 772–791, Dec. 2021, doi: 10.1016/j.ejor.2021.03.009.

[8] S. Butenko, "Maximum independent set and related problems, with applications," Ph.D. dissertation, Ind. Syst. Eng. Dept., Univ. Florida, Gainesville, FL, USA, 2003. [Online]. Available: https://ufdcimages.uflib.ufl.edu/UF/E0/00/10/11/00001/butenko_s.pdf

[9] V. Boginski, S. Butenko, and P. M. Pardalos, "Network-based techniques in the analysis of the stock market," in *Supply Chain and Finance*, P. M. Pardalos, A. Migdalas, and G. Baourakis, Eds. Singapore: World Scientific, 2004, pp. 1–14, doi: 10.1142/9789812562586_0001.

[10] M. Marzec, "Portfolio optimization: Applications in quantum computing," in *Handbook of High-Frequency Trading and Modeling in Finance*, I. Florescu, M. C. Mariani, H. E. Stanley, and F. G. Viens, Eds. Hoboken, NJ, USA: Wiley, 2016, pp. 73–106, doi: 10.1002/9781118593486.ch4.

[11] C. Bick, E. Gross, H. A. Harrington, and M. T. Schaub, "What are higher-order networks?" *SIAM Rev.*, vol. 65, no. 3, pp. 686–731, Aug. 2023, doi: 10.1137/21M1414024.

[12] F. Battiston, G. Cencetti, I. Iacopini, V. Latora, M. Lucas, A. Patania, J.-G. Young, and G. Petri, "Networks beyond pairwise interactions: Structure and dynamics," *Phys. Rep.*, vol. 874, pp. 1–92, Aug. 2020, doi: 10.1016/j.physrep.2020.05.004.

[13] S. Saha, J. Gao, and R. Gerlach, "A survey of the application of graph-based approaches in stock market analysis and prediction," *Int. J. Data Sci. Analytics*, vol. 14, no. 1, pp. 1–15, Jun. 2022, doi: 10.1007/s41060-021-00306-9.

[14] Y. Yan, B. Wu, T. Tian, and H. Zhang, "Development of stock networks using part mutual information and Australian stock market data," *Entropy*, vol. 22, no. 7, p. 773, Jul. 2020, doi: 10.3390/e22070773.

[15] A. Bielinskyi, V. Soloviev, S. Hushko, A. Kiv, and A. Matviychuk, "High-order networks and stock market crashes," in *Proc. 10th Int. Conf. Monitor., Model. Manage. Emergent Economy*, 2022, pp. 134–144, doi: 10.5220/0011931900003432.

[16] A. Lucas, "Ising formulations of many NP problems," *Frontiers Phys.*, vol. 2, pp. 1–15, Feb. 2014, Art. no. 5, doi: 10.3389/fphy.2014.00005.

[17] H. Goto, K. Tatsumura, and A. R. Dixon, "Combinatorial optimization by simulating adiabatic bifurcations in nonlinear Hamiltonian systems," *Sci. Adv.*, vol. 5, no. 4, Apr. 2019, Art. no. eaav2372, doi: 10.1126/sciadv.aav2372.

[18] K. Tatsumura, A. R. Dixon, and H. Goto, "FPGA-based simulated bifurcation machine," in *Proc. 29th Int. Conf. Field Program. Log. Appl. (FPL)*, Sep. 2019, pp. 59–66, doi: 10.1109/FPL.2019.00019.

[19] H. Goto, K. Endo, M. Suzuki, Y. Sakai, T. Kanao, Y. Hamakawa, R. Hidaka, M. Yamasaki, and K. Tatsumura, "High-performance combinatorial optimization based on classical mechanics," *Sci. Adv.*, vol. 7, no. 6, Feb. 2021, Art. no. eabe7953, doi: 10.1126/sciadv.abe7953.

[20] K. Tatsumura, M. Yamasaki, and H. Goto, "Scaling out Ising machines using a multi-chip architecture for simulated bifurcation," *Nature Electron.*, vol. 4, no. 3, pp. 208–217, Mar. 2021, doi: 10.1038/s41928-021-00546-4.

[21] T. Kanao and H. Goto, "Simulated bifurcation for higher-order cost functions," *Appl. Phys. Exp.*, vol. 16, no. 1, Jan. 2023, Art. no. 014501, doi: 10.35848/1882-0786/acaba9.

[22] M. W. Johnson et al., "Quantum annealing with manufactured spins," *Nature*, vol. 473, no. 7346, pp. 194–198, May 2011, doi: 10.1038/nature10012.

[23] A. D. King et al., "Quantum critical dynamics in a 5,000-qubit programmable spin glass," *Nature*, vol. 617, no. 7959, pp. 61–66, May 2023, doi: 10.1038/s41586-023-05867-2.

[24] T. Honjo, T. Sonobe, K. Inaba, T. Inagaki, T. Ikuta, Y. Yamada, T. Kazama, K. Enbutsu, T. Umeki, R. Kasahara, K.-I. Kawarabayashi, and H. Takesue, "100,000-spin coherent Ising machine," *Sci. Adv.*, vol. 7, no. 40, Oct. 2021, Art. no. eabh0952, doi: 10.1126/sciadv.abh0952.

[25] D. Pierangeli, G. Marcucci, and C. Conti, "Large-scale photonic Ising machine by spatial light modulation," *Phys. Rev. Lett.*, vol. 122, no. 21, May 2019, Art. no. 213902, doi: 10.1103/PhysRevLett.122.213902.

[26] F. Cai, S. Kumar, T. Van Vaerenbergh, X. Sheng, R. Liu, C. Li, Z. Liu, M. Foltin, S. Yu, Q. Xia, J. J. Yang, R. Beausoleil, W. D. Lu, and J. P. Strachan, "Power-efficient combinatorial optimization using intrinsic noise in memristor Hopfield neural networks," *Nature Electron.*, vol. 3, no. 7, pp. 409–418, Jul. 2020, doi: 10.1038/s41928-020-0436-6.

[27] N. A. Aadit, A. Grimaldi, M. Carpentieri, L. Theogarajan, J. M. Martinis, G. Finocchio, and K. Y. Camsari, "Massively parallel probabilistic computing with sparse Ising machines," *Nature Electron.*, vol. 5, no. 7, pp. 460–468, Jun. 2022, doi: 10.1038/s41928-022-00774-2.

[28] W. Moy, I. Ahmed, P.-W. Chiu, J. Moy, S. S. Sapatnekar, and C. H. Kim, "A 1,968-node coupled ring oscillator circuit for combinatorial optimization problem solving," *Nature Electron.*, vol. 5, no. 5, pp. 310–317, May 2022, doi: 10.1038/s41928-022-00749-3.

[29] A. Sharma, R. Afoakwa, Z. Ignjatovic, and M. Huang, "Increasing Ising machine capacity with multi-chip architectures," in *Proc. 49th Annu. Int. Symp. Comput. Archit.*, Jun. 2022, pp. 508–521, doi: 10.1145/3470496.3527414.

[30] T. Takemoto, M. Hayashi, C. Yoshimura, and M. Yamaoka, "A 2 × 30k-spin multi-chip scalable CMOS annealing processor based on a processing-in-memory approach for solving large-scale combinatorial optimization problems," *IEEE J. Solid-State Circuits*, vol. 55, no. 1, pp. 145–156, Jan. 2020, doi: 10.1109/JSSC.2019.2949230.

[31] K. Kawamura, J. Yu, D. Okonogi, S. Jimbo, G. Inoue, A. Hyodo, Á. L. García-Anas, K. Ando, B. H. Fukushima-Kimura, R. Yasudo, T. Van Chu, and M. Motomura, "Amorphica: 4-Replica 512 fully connected spin 336 MHz metamorphic annealer with programmable optimization strategy and compressed-spin-transfer multi-chip extension," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2023, pp. 42–44, doi: 10.1109/ISSCC42615.2023.10067504.

[32] S. Matsubara, M. Takatsu, T. Miyazawa, T. Shibasaki, Y. Watanabe, K. Takemoto, and H. Tamura, "Digital annealer for high-speed solving of combinatorial optimization problems and its applications," in *Proc. 25th Asia South Pacific Design Autom. Conf. (ASP-DAC)*, Jan. 2020, pp. 667–672, doi: 10.1109/ASP-DAC47756.2020.9045100.

[33] H. M. Waidyasooriya and M. Hariyama, "Highly-parallel FPGA accelerator for simulated quantum annealing," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 4, pp. 2019–2029, Oct. 2021, doi: 10.1109/TETC.2019.2957177.

[34] T. Okuyama, T. Sonobe, K.-I. Kawarabayashi, and M. Yamaoka, "Binary optimization by momentum annealing," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 100, no. 1, Jul. 2019, Art. no. 012111, doi: 10.1103/PhysRevE.100.012111.

[35] F. Barahona, "On the computational complexity of Ising spin glass models," *J. Phys. A, Math. Gen.*, vol. 15, no. 10, pp. 3241–3253, Oct. 1982, doi: 10.1088/0305-4470/15/10/028.

[36] S. Yoo, H. Kim, J. Kim, S. Park, J.-Y. Kim, and J. Oh, "LightTrader: A standalone high-frequency trading system with deep learning inference accelerators and proactive scheduler," in *Proc. IEEE Int. Symp. High-Perform. Comput. Archit. (HPCA)*, Feb. 2023, pp. 1017–1030, doi: 10.1109/HPCA56546.2023.10070930.

[37] M. Fil and L. Kristoufek, "Pairs trading in cryptocurrency markets," *IEEE Access*, vol. 8, pp. 172644–172651, 2020, doi: 10.1109/ACCESS.2020.3024619.

[38] B. Huang, Y. Huan, L. D. Xu, L. Zheng, and Z. Zou, "Automated trading systems statistical and machine learning methods and hardware implementation: A survey," *Enterprise Inf. Syst.*, vol. 13, no. 1, pp. 132–144, Jan. 2019, doi: 10.1080/17517575.2018.1493145.

[39] S. Denholm, H. Inoue, T. Takenaka, T. Becker, and W. Luk, "Network-level FPGA acceleration of low latency market data feed arbitration," *IEICE Trans. Inf. Syst.*, vol. 98, no. 2, pp. 288–297, 2015, doi: 10.1587/transinf.2014RCP0011.

[40] C. Leber, B. Geib, and H. Litz, "High frequency trading acceleration using FPGAs," in *Proc. 21st Int. Conf. Field Program. Log. Appl.*, Sep. 2011, pp. 317–322, doi: 10.1109/FPL.2011.64.

[41] L. Malceniece, K. Malcenieks, and T. J. Putninš, "High frequency trading and comovement in financial markets," *J. Financial Econ.*, vol. 134, no. 2, pp. 381–399, Nov. 2019, doi: 10.1016/j.jfineco.2018.02.015.

[42] J. Brogaard, T. Hendershott, and R. Riordan, "High-frequency trading and price discovery," *Rev. Financial Stud.*, vol. 27, no. 8, pp. 2267–2306, Aug. 2014, doi: 10.1093/rfs/hhu032.

[43] S. Spyrou, "Herding in financial markets: A review of the literature," *Rev. Behav. Finance*, vol. 5, no. 2, pp. 175–194, Nov. 2013, doi: 10.1108/RBF-02-2013-0009.

[44] K. Tatsumura, R. Hidaka, M. Yamasaki, Y. Sakai, and H. Goto, "A currency arbitrage machine based on the simulated bifurcation algorithm for ultrafast detection of optimal opportunity," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Oct. 2020, pp. 1–5, doi: 10.1109/ISCAS45731.2020.9181114.

[45] H. Goto, "Bifurcation-based adiabatic quantum computation with a nonlinear oscillator network," *Sci. Rep.*, vol. 6, no. 1, Feb. 2016, Art. no. 21686, doi: 10.1038/srep21686.

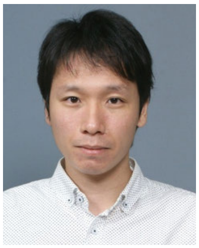
[46] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-26, no. 1, pp. 43–49, Feb. 1978, doi: 10.1109/TASSP.1978.1163055.

[47] G. Marsaglia, "Xorshift RNGs," *J. Stat. Softw.*, vol. 8, no. 14, pp. 1–6, 2003, doi: 10.18637/jss.v008.i14.

[48] W. F. Sharpe, "Mutual fund performance," *J. Bus.*, vol. 39, no. 1, pp. 119–138, Jan. 1966. [Online]. Available: https://www.jstor.org/stable/2351741

[49] D. K. Backus, A. W. Gregory, and C. I. Telmer, "Accounting for forward rates in markets for foreign currency," *J. Finance*, vol. 48, no. 5, pp. 1887–1908, Dec. 1993, doi: 10.1111/j.1540-6261.1993.tb05132.x.

KOSUKE TATSUMURA (Member, IEEE) received the B.E., M.E., and Ph.D. degrees in electronics, information and communications engineering from Waseda University, Japan, in 2000, 2001, and 2004, respectively. After working as a Postdoctoral Fellow with Waseda University, he joined Toshiba Corporation, in 2006. He is currently a Chief Research Scientist, leading a research team and several projects toward realizing innovative industrial systems based on cutting-edge computing technology. He was a member of the Emerging Research Devices (ERD) Committee in the International Technology Roadmap for Semiconductors (ITRS), from 2013 to 2015. He has been a Lecturer with Waseda University, since 2013. He was a Visiting Researcher with the University of Toronto, from 2015 to 2016. He received the Best Paper Award at IEEE International Conference on Field-Programmable Technology (FTP), in 2016. His research interests include domain-specific computing, quantum/quantum-inspired computing, and their applications.



RYO HIDAKA received the B.E. and M.E. degrees in systems design and informatics from the Kyushu Institute of Technology, Japan, in 2006 and 2008, respectively. He joined Toshiba Corporation, in 2008. He was engaged in the development of main processors (2D-to-3D conversion and local dimming) for digital televisions, an image recognition processor called Visconti^{TM}, and host controllers for flash-memory cards. His current research interests include domain-specific computing, high-level synthesis design methodology, and the proof-of-concept study with FPGA devices.

JUN NAKAYAMA received the Bachelor of Arts degree in economic and social studies from The University of Manchester, U.K., in 2008, and the Master of Business Administration (M.B.A.) degree in finance from Hitotsubashi University, Japan, in 2017. He was a Portfolio Manager of Nomura Asset Management Company Ltd., from 2008 to 2020. He was engaged in the development and management of quant-based funds. He joined the Corporate Research and Development Center, Toshiba Corporation, in 2020. He is also currently pursuing the Ph.D. degree with the Financial Strategy Program, Hitotsubashi University Business School. His research interests include quantitative investment strategies, quantum-inspired computing technology, and trading strategies with advanced technologies.

TOMOYA KASHIMATA received the B.E. and M.E. degrees in computer science and engineering from Waseda University, Japan, in 2018 and 2020, respectively. He joined the Corporate Research and Development Center, Toshiba Corporation, Japan, in 2020. His research interests include computer architecture, reconfigurable architecture, and processor in memory.

MASAYA YAMASAKI received the B.E. and M.E. degrees in computer science and communication engineering from Kyushu University, Japan, in 1997 and 1999, respectively. He joined Toshiba Corporation, in 1999. He was engaged in the development of image processing engines (interframe interpolation technology) for digital televisions (including ones with Cell Broadband Engines) and FPGA-based coprocessors for multi-channel video recording, three-dimensional display, and industrial systems. His research interests include domain-specific computing, high-level synthesis design space exploration, and the proof-of-concept study with FPGA devices.