

Received 25 September 2023, accepted 14 October 2023, date of publication 23 October 2023, date of current version 1 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3326816

RESEARCH ARTICLE

Real-Time Trading System Based on Selections of Potentially Profitable, Uncorrelated, and Balanced Stocks by NP-Hard Combinatorial Optimization

KOSUKE TATSUMURA¹, (Member, IEEE), RYO HIDAKA¹, JUN NAKAYAMA¹,
TOMOYA KASHIMATA¹, AND MASAYA YAMASAKI¹

Corporate Research and Development Center, Toshiba Corporation, Kawasaki 212-8582, Japan

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

ABSTRACT Financial portfolio construction problems are often formulated as quadratic and discrete (combinatorial) optimization that belong to the nondeterministic polynomial time (NP)-hard class in computational complexity theory. Ising machines are hardware devices that work in quantum-mechanical/quantum-inspired principles for quickly solving NP-hard optimization problems, which potentially enable making trading decisions based on NP-hard optimization in the time constraints for high-speed trading strategies. Here we report a real-time stock trading system that determines long(buying)/short(selling) positions through NP-hard portfolio optimization for improving the Sharpe ratio using an embedded Ising machine based on a quantum-inspired algorithm called simulated bifurcation. The Ising machine selects a balanced (delta-neutral) group of stocks from an N -stock universe according to an objective function involving maximizing instantaneous expected returns defined as deviations from volume-weighted average prices and minimizing the summation of statistical correlation factors (for diversification). It has been demonstrated in the Tokyo Stock Exchange that the trading strategy based on NP-hard portfolio optimization for $N = 128$ is executable with the FPGA (field-programmable gate array)-based trading system with a response latency of 164 μ s.

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

I. INTRODUCTION

Many portfolio construction/selection problems in finance are, with considering minimum transaction lots or other discreteness of decision variables as realistic constraints, known to be nondeterministic polynomial (NP)-hard in computer science [1], [2]. Those include discrete optimizations of Markowitz's mean-variance model [3] for better risk-return characteristics [4], [5], multi-period portfolio optimizations (or optimal trading trajectory problems) [6], [7], [8], and correlation-diversified portfolio constructions

including maximum independent set (MIS) problem-based ones [9], [10], [11] and permutation of assets-based one [12].

Recently, special-purpose computers for NP-hard combinatorial (or discrete) optimization, called Ising machines [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], have attracted intense attention. Ising problems are the ground (energy minimum)-state search problems of Ising spin models [31], which consist of binary variables, called spins, coupled each other with pairwise interactions (two-state interacting variables like the Ising model have been utilized for analyzing various physical or social systems [32], [33], [34]). The Ising problem belongs to the NP-hard class [35], [36]; a variety of notoriously hard problems can be represented in the form of the Ising

The associate editor coordinating the review of this manuscript and approving it for publication was Yiming Huo¹.

problem [36], including discrete portfolio optimization [4], [5], [6], [7], [8] and market graph analysis [9], [10], [11], [37], [38] in finance. The Ising machine is a heuristic methodology and quickly searches for the optimal (exact) or near-optimal solutions of the Ising problem in the whole solution space. Many Ising machines have claimed higher speed performance than simulated annealing [39], [40] (on von Neumann computers), a conventional heuristic for combinatorial optimization. The generalization of the Ising machine (quadratic) to higher-order problems [17], [41] has also been studied to analysis higher-order networks [42], [43], [44], [45], [46], [47].

The Ising machines are implemented with various hardware including superconducting flux qubits [18], [19], hybrid electronic-optical systems [20], [21], memristor-based neural networks [22], probabilistic bits [23], coupled-oscillator circuits [24], analog computing units [25], application specific integrated circuits (ASICs) [26], [27], [28], field programmable gate arrays (FPGAs) [13], [14], [15], [16], [29], and graphics processing units (GPUs) [13], [15], [30].

The Ising machines may enable making more rational judgments based on NP-hard combinatorial optimizations for automated trading systems [48], [49], [50], [51], [52] that become increasingly important in financial markets [55], [56]. Those trading systems are typical real-time systems that must respond (sense, judge, and react) within critically defined time constraints. Many high-speed trading systems [48], [49], [50], [51], [52] utilize FPGAs to shorten the latency from the market feed arrival to order packet issuance. Thus, among various Ising machines, FPGA-based ones (Ising machines that can be accelerated with modern FPGA architectures [53]) are suitable for high-speed trading systems because they can be embedded together with other system components in the FPGAs. High-speed trading strategies based on combinatorial/discrete optimization and the trading systems utilizing FPGA-based embeddable Ising machines as in [37] and [38] have been, however, not extensively studied. The execution capability of such a trading system needs to be validated in the actual market because the duration times of the trading opportunities of a strategy (the time constrains for the system) are determined by the activities of other trading entities.

Here we propose a trading strategy based on selections of potentially profitable, uncorrelated, and balanced stocks by NP-hard combinatorial optimization and show through real-time trading that the strategy is executable with an automated real-time system using an FPGA-based embedded Ising machine for the discrete selection problem.

Based on the demand in the direction of convergence of the stock price to the volume-weighted average price (VWAP), the proposed strategy considers the deviations of stock prices from the VWAPs as instantaneous expected returns and selects a balanced (delta-neutral) group of stocks from an N -stock universe according to an objective function involving maximizing the expected returns and minimizing the

summation of statistical correlation factors (for correlation-diversification). The selection problem is formulated as quadratic and discrete optimization and solved by an Ising machine based on a quantum-inspired algorithm called simulated bifurcation (SB) [13], [14], [15], [16], [17]. SB was derived from a classical counterpart to a quantum adiabatic optimization method called a quantum bifurcation machine [54] and numerically simulates the adiabatic time-evolution of a classical nonlinear oscillator network exhibiting bifurcation phenomena, where two branches of the bifurcation in each oscillator correspond to two states of each Ising spin. The algorithm of SB is highly parallelizable and thus can be accelerated with parallel processors such as FPGAs (field-programmable gate arrays) [14]. We use an FPGA-based SB machine (SBM, a hardware implementation of SB) because it can be integrated in an FPGA together with other system components to shorten the system-wide latency. To further reduce the system-wide latency by decreasing the input data size of the SB machine from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$, we separate the data describing the problem into two components that change tick-by-tick or day-by-day and customize the basic SBM design [14]. We discuss the execution capability of the system by comparing 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.

II. COMPOSITION OF SECTIONS

This paper is multidisciplinary and across several fields including financial engineering, discrete mathematics, and computer engineering. The rest of the paper is organized as follows. In Sec. III (trading strategy), we describe the proposed strategy and formulate the discrete selection problem in the forms of quadratic unconstrained binary optimization (QUBO) problem and the Ising problem. Sec. IV (system) describes the architecture of the system, the customization of the SBM core circuit, and the implementation details. Sec. V (experiment) describes the transaction records in the TSE and the execution capability of the system. Sec. VI concludes the paper.

III. TRADING STRATEGY

A. DISCRETE OPTIMIZATION-BASED STRATEGY

The proposed strategy considers the deviations of stock prices from the VWAPs as instantaneous expected returns and bets that the deviations would eventually converge (partially) in the trading hours. To improve the reward-to-variability ratio (or the Sharpe ratio [57], [58]), it simultaneously holds multiple positions selected through a discrete portfolio optimization problem making the group of positions being market-neutral [59] and correlation-diversified [9], [10], [11], [12].

There is demand in the direction of convergence of the stock price to the VWAP [60], [61], [62]. For institutional investors mainly through passive investments, one of the

common methods for reducing the trading impact on market prices is that the fund managers, with a certain fee promised, ask brokerages to execute their large volume trades on the VWAP determined at the end of the trading hours. If the average executed price is the same as the end-of-trading-hours VWAP, the brokerage earns the fee. If the brokerage executes the trades at prices more favorable than the VWAP, this brokerage earns more than the fee.

Considering the deviations of stock prices from the VWAP as expected returns, the strategy takes long positions of the underperforming stocks and short positions of the outperforming stocks and statistically expects that the underperforming stocks would move up while the outperforming stocks would move down. To adapt to various market conditions (uptrend, downtrend, or sideways), the strategy matches long(/short) positions with short(/long) positions so that the overall deltas of the positions total almost zero (delta neutral) [59]. In addition, to statistically reduce the deviation of the returns (risk), the strategy incorporates the concept of correlation-diversified portfolio [9], [10], [11], [12]; the multiple long/short positions are selected so that the stocks involved are uncorrelated with each other.

The Sharpe ratio [57] 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 [58]. To enhance the Sharpe ratio of the proposed strategy, a group including N_s stocks is selected from an N -stock universe as the candidates of open positions (positions to be taken) so that (i) the summation of instantaneous expected returns is maximized (for maximizing returns), (ii) the summation of statistical correlation factors is minimized (for diversification), and (iii) the numbers of long/short positions are equal (delta-neutral). This is a discrete optimization problem. The selection of N_s -stock group is executed every time the market situation changes and then the selected group is evaluated for determining the opening.

The deviation of the stock price from the VWAP (Δp_i) normalized with the base price on the day (p_i^b) is expressed by $\Delta p_i = (p_i - VWAP_i)/p_i^b$, where p_i is the middle price between the best ask (*ask*) and the best bid (*bid*). When the sign of Δp_i is negative (/positive), i th stock is the candidate of long (/short) position. The absolute value of Δp_i corresponds to the instantaneous expected return of the i th-stock position.

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); $L_i = \lfloor A_{\text{trans}}/S_i^{\text{min}}p_i^b \rfloor$. 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 positions are not allowed.

In this work, the correlation factor between i th and j th stocks for a business day ($\hat{\sigma}_{i,j}$) is defined based on the price

deviation sequences against the VWAP as follows.

$$\hat{\sigma}_{i,j} = \frac{\sum_k (p_i^k - VWAP_i^k) (p_j^k - VWAP_j^k)}{\sum_k |p_i^k - VWAP_i^k| \sum_k |p_j^k - VWAP_j^k|}, \quad (1)$$

where p_i^k and $VWAP_i^k$ are the middle price and VWAP of i th stock sampled at one-second intervals. The correlation factor ($\sigma_{i,j}$) in the strategy is the average value for the last five business days of $\hat{\sigma}_{i,j}$ and is normalized to be in $[0, 1]$.

B. FORMULATION

The problem to select N_s stocks from an N -stock universe according to a cost (objective) function involving maximizing instantaneous expected returns ($|\Delta p_i|$) and minimizing the summation of statistical correlation factors ($\sigma_{i,j}$) of the involved stocks under the constrain for delta-neutral positions is formulated in the form of quadratic unconstrained binary optimization (QUBO).

The formulation of the QUBO is in a one-to-one relationship with the Ising model [31] that the Ising machine takes as input data (see APPENDIX A for the mutual conversions). Both the QUBO and Ising formulations describe the system under interest using two-state (binary) variables which have pairwise interactions (second-order interactions) each other, where the two-state variables are bits ($b_i \in \{0, 1\}$) in the QUBO and spins ($s_i \in \{-1, 1\}$) in the Ising model. The dynamics or time-evolution of the two-state interacting variables describing physical or social systems according to variously-defined updating rules of the two-state variables (corresponding to the updating rules in cellular automata) have been studied from multiple perspectives including phase transition, percolation, and optimization [32], [33], [34]. For examples in sociology, majority-voting rules are used for modeling the diffusion of infection and rumor [33], [34]. Simulated bifurcation [13], [15], utilized in this work for discrete optimization, updates the states of oscillators (corresponding to the spin variables) depending on the joint pairwise interactions in an N -dimensional space where the energy landscape gradually changes (bifurcation happens).

In this subsection, we explain the QUBO formulation of the N_s -stock selection problem for explanatory clarity, and then mention the Ising formulation in the next subsection. The primitive data defining the problem ($\{\Delta p_i\}$ vector and σ matrix) are converted directly to the Ising formulation in the trading system (not via the QUBO formulation).

Define a decision (bit) variable b_i ($b_i \in \{0, 1\}$) as taking value 1 if i th stock is selected and 0 otherwise. When i th stock is selected, the sign of Δp_i [$\text{sgn}(\Delta p_i)$] indicates whether it corresponds to a long or short position. We prepare N bit variables for an N -stock universe.

In the QUBO formulation, we search for the bit configuration $\{b_i\}$ that minimizes the QUBO cost function H_{QUBO} .

H_{QUBO} is a linear combination of a cost function H_{cost} and a penalty function H_{penalty} .

$$H_{\text{QUBO}} = \sum_i^N \sum_j^N Q_{i,j} b_i b_j = H_{\text{cost}} + H_{\text{penalty}}. \quad (2)$$

The cost function to be minimized is defined by

$$H_{\text{cost}} = \sum_i \sum_j Q_{i,j}^{\text{cost}} b_i b_j, \quad (3)$$

$$Q_{i,j}^{\text{cost}} = \begin{cases} -c_1 |\Delta p_i| & (\text{if } i = j), \\ \sigma_{i,j} & (\text{otherwise}), \end{cases} \quad (4)$$

where c_1 is a positive coefficient. Note that $b_i^2 = b_i$ for diagonal terms ($i = j$). The constraints for N_s -stock selection and delta-neutral positions are represented as a penalty function expressed by

$$H_{\text{penalty}} = c_2 \left(\left(\sum_i b_i \right) - N_s \right)^2 + c_3 \left(\sum_i \text{sgn}(\Delta p_i) b_i \right)^2. \quad (5)$$

where c_2 and c_3 are positive coefficients. The first and second terms correspond to the constraints for N_s -stock selection and delta-neutral positions, respectively. Constraint violations increase the penalty, with $H_{\text{penalty}} = 0$ if there are no violations. Note that the nondiagonal elements in the coupling coefficient matrix Q in Eq. (2) include not only $\sigma_{i,j}$ in Eq. (4) but also components of $\text{sgn}(\Delta p_i)$ coming from the second term in Eq. (5). QUBOs are known to be NP-hard problems for classical computers [36]. Since the cost function in Eq. (4) is quadratic, the discrete optimization involved in the strategy is thought to be NP-hard problems.

C. SEPARATION OF PROBLEM COMPONENTS

The discrete optimization problem to be solved at a market situation is described as an $N \times N$ size of coupling coefficient matrix Q (in Eq. (2)), which should be transferred to the Ising machine (in this work, SBM) every time the market situation changes. To reduce the system-wide latency by decreasing the size of data transferred from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$, we separate the data describing the problem into two components that change tick-by-tick or day-by-day. We prepare additional circuit units for the computation depending only on the tick-by-tick data to the basic SBM design (see Sec. IV).

In the QUBO formulation in Eqs. (2), (3), (4) and (5), the $\{\Delta p_i\}$ vector is the tick-by-tick change component and the σ matrix is the day-by-day change component. The QUBO problem can be represented in the form of the Ising problem (see APPENDIX A), where the decision variables are spins s_i ($s_i \in \{-1, 1\}$) and the problem is represented by a coupling coefficient matrix J and a bias vector h . We describe the Ising cost function H_{Ising} as a linear combination of terms that include the day-by-day change components ($J^{\text{day}}, h^{\text{day}}$) or tick-by-tick change components ($J^{\text{tick}}, h^{\text{tick}}$)

as follows;

$$H_{\text{Ising}} = -\frac{1}{2} \sum_i^N \sum_j^N J_{i,j}^{\text{day}} s_i s_j - \frac{1}{2} \sum_i^N \sum_j^N J_{i,j}^{\text{tick}} s_i s_j + \sum_i^N h_i^{\text{day}} s_i + \sum_i^N h_i^{\text{tick}} s_i. \quad (6)$$

Here, the ($J^{\text{day}}, h^{\text{day}}$) and ($J^{\text{tick}}, h^{\text{tick}}$) can be calculated from the σ matrix and $\{\Delta p_i\}$ vector, respectively.

The SBM core described in the next section stores the ($J^{\text{day}}, h^{\text{day}}$) and ($J^{\text{tick}}, h^{\text{tick}}$) data in the separated memories. Note that the size of $J_{i,j}^{\text{tick}}$ is $N \times N$, but the N -size intermediate values are stored in the separated memory (see APPENDIX B for details). When a market feed (informing the change of *ask* or *bid* of a stock) arrives, the SBM core updates the ($J^{\text{tick}}, h^{\text{tick}}$) intermediate data [the size is $\mathcal{O}(N)$] with keeping the ($J^{\text{day}}, h^{\text{day}}$) data [the size is $\mathcal{O}(N^2)$].

IV. SYSTEM

The real-time stock trading system is a hybrid FPGA/CPU system, featuring an event-driven SBM module that starts processing the discrete optimization involved in the proposed strategy when detecting the changes in *ask* or *bid* of tradable stocks. The system-wide latency from the market feed arrival to order packet issuance is shortened by co-integrating, in the FPGA, the SBM module together with other system components including communication interfaces. The processing units and memory subsystems in the basic SBM circuit design [14] have been customized (modified) for the proposed strategy to further improve the system latency.

A. ARCHITECTURE

Figure 1 (a) and (b) show the block diagram and timing chart of the hybrid FPGA/CPU system. The FPGA part responds to the changes in the market in a low latency, i.e., it receives the market information, determines the opening of positions based on the NP-hard portfolio optimization by the SBM module, and then issues the order packets. The CPU part controls the whole system and manages the positions using state machines for opened positions (the closing of the positions is determined by the CPU part). 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.

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* or *bid* for the N -stock universe, the SBM module including the three memory units ($\Delta p, \sigma, VWAP$) which are updated at different timing, a judgment module, a message generator, and a transmitter (TX). Those components are implemented as independent (not synchronized) circuit modules, which are

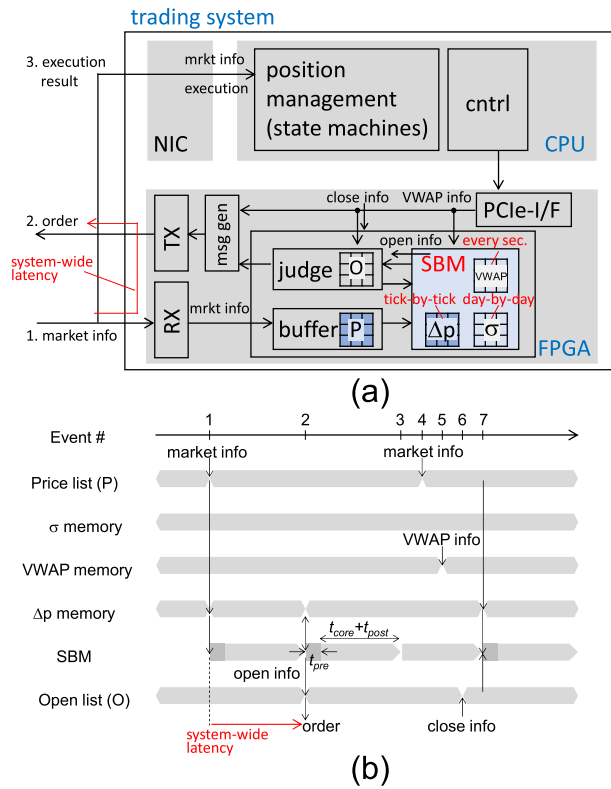


FIGURE 1. System architecture (a hybrid FPGA/CPU system). (a) Block diagram. (b) Timing chart.

connected with directed streaming data channels with FIFO (first-in-first-out) buffers.

One of the characteristics of the SBM module is that it has three memory units (Δp , σ , and $VWAP$) to store data updated at the three timing of tick-by-tick, day-by-day, and every second. The $VWAP$ information, a $\{VWAP_i\}$ vector, is updated in the CPU part and informed to the FPGA part at one-second intervals. The SBM module also has a preprocessing submodule (pre) to generate the Ising problem described by (J^{day}, h^{day}) and (J^{tick}, h^{tick}) based on the data in the three memory units. The data in the Δp memory is changed depending also on the open list (O memory) in the judgment module. The open position is registered in the open list when the opening is decided (before the issuance of the order) and deregistered from there when the closing of the position is confirmed with the message from the CPU part. The Δp of the stocks listed in the O memory is set to zero as the duplicate opening is prohibited.

Figure 1 (b) shows the timing chart for the operation of the SBM module when representative events (Events 1 to 7) happen. When no event happens for a certain time, the SBM module is idling (polling to the FIFO buffers from the price buffer, judgment, and PCIe I/F modules). When a market feed arrives (Event 1), the SBM module immediately starts the preprocessing (updating of the Δp memory and the J^{tick}/h^{tick} memory) and then the main processing (the discrete optimization). After that, the SBM module evaluates the

solution output from the core circuit in terms of the constraint violation and the objective function and then informs the open candidates to the judgment module if the evaluation passes (postprocessing, $post$). After checking the open candidates, the judgment module finally determines the open positions, registers them in the O memory, and concurrently issues order packets via the message generator (Event 2).

Regardless of whether or not order packets are issued, the SBM module repeats the main processing for a predetermined number of times with different initial states generated by an internal random number generator [63]. As the simulated bifurcation is a heuristic algorithm, the SBM module may find a better solution or another solution enough for the opening. When repeating the main processing, the SBM module also repeats the preprocessing if the order packets have been issued at the last run (in the case of Event 2) since the Δp memory (O memory) has been changed, but skips the preprocessing otherwise (Event 3). When the Ising problem changes during the main processing because of the arrivals of the new market feed (Event 4), the $VWAP$ updating information (Event 5), or the close confirmation information (Event 6), the information is incorporated at the beginning of the next execution of the SBM module (Event 7).

B. CUSTOMIZED SBM CORE CIRCUIT

To reduce the data size transferred to the SBM core from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$ (for improving the system latency) when the market situation (or the internal state) changes, we prepare additional computation and memory units to the basic SBM design. Instead of combining the tick-by-tick (N -size) and day-by-day ($N \times N$ -size) change information as the coefficients describing the Ising problem, we store that information in separated memory units, separately calculate the updating components of internal variables (corrections of momenta) coming from those two components and then combine the updating components (the number of the components is N).

Simulated bifurcation [13], [15] simulates the time evolution of N nonlinear oscillators according to the Hamiltonian equations of motion, where the nonlinear oscillators correspond to the spin variables and the state of i th oscillator is described by the position and momentum (x_i, y_i) . The SB time evolution step consists of calculating the correction of momenta $\{\Delta y_i\}$ based on the many-body interaction [computationally corresponding to the matrix-vector multiplication (MM) of the J coupling matrix and $\{x_i\}$ position vector] and calculating the updated (time-evolved) state variables, $\{x_i^{k+1}\}$ and $\{y_i^{k+1}\}$, from the $\{\Delta y_i\}$, $\{h_i\}$, and the current state variables, $\{x_i^k\}$ and $\{y_i^k\}$. The additional circuits calculate the correction of the momentum depending only on the tick-by-tick J and h components (Δy_i^{tick}).

Figure 2 shows the block diagram of the SB core circuit, where the additional circuit units for the computation depending only on the tick-by-tick change problem components (J^{tick}, h^{tick}) are blue highlighted and the remaining units are

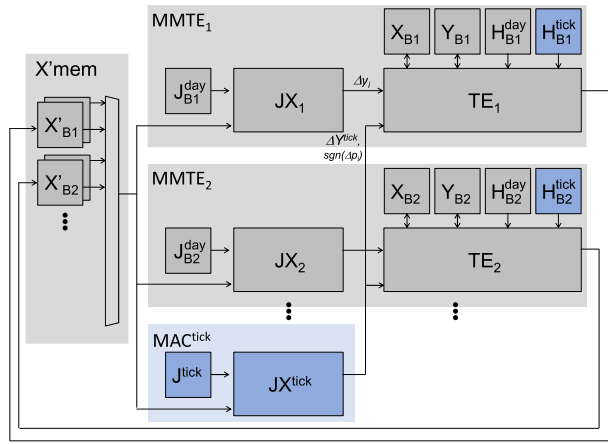


FIGURE 2. Circuit architecture of the SBM core. To shorten the data input time, additional circuit units (blue highlighted) for the computation depending on the tick-by-tick data are introduced to the basic SBM design [14].

architecturally the same as in the basic SBM design [14]. In the basic design, the main computation components are JX units corresponding to the multiply-accumulate (MAC) operations of $\sum_{j=1}^N J_{ij}x_j$ and TE units corresponding to the time-evolution operation, which are combined to be MMTE units (each responsible for updating a subgroup of coupled oscillators). The MMTE units are organized with the global X'_{mem} memory unit to make a circulative structure as a whole corresponding to the iteration of the SB time-evolution steps.

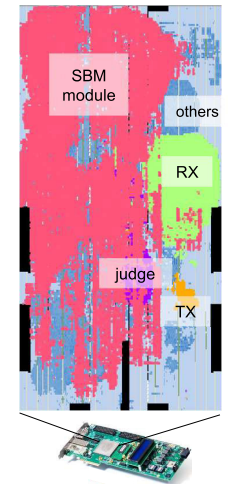
The J^{tick} and h^{tick} data are stored in the J^{tick} and H^{tick} memory units, which are separated from the J^{day} and H^{day} memory units storing the day-by-day change problem components (J^{day} , h^{day}). The JX^{tick} module calculates an intermediate value ΔY^{tick} common for all the oscillators and supply the ΔY^{tick} and $\text{sgn}(\Delta p_i)$ data to the TE units. The TE unit calculates the Δy_i^{tick} for each oscillator based on the ΔY^{tick} , $\text{sgn}(\Delta p_i)$, x_i and h_i and updates the state of i th oscillator with the Δy_i^{tick} . See APPENDIX B for details.

C. IMPLEMENTATION

We implemented the system described in Sec. IV-A with a CPU server with a network interface card (NIC) and an FPGA board having another network interface (see APPENDIX C for details).

Figure 3 (a) shows the architecture and implementation results of the SBM module for 128-stock universes ($N = 128$). Among three variants of simulated bifurcation (adiabatic, ballistic, and discrete SBs) [15], ballistic SB is adopted in this work, with the SB parameters of $N_{\text{step}}=300$ (the number of SB time evolution steps per execution) and $dt = 0.02$ (the integral time step). The machine size (the number of spins) is 128 spins with all-to-all connectively, and the computation precision is 32-bit floating point. Figure 3 (b) shows the result of the placement of system modules in the FPGA. The SBM module is dominant, and the circuit resources used are listed in Fig. 3 (a). The system clock frequency determined as a result of

	SBM module
Architecture	
algorithm version	ballistic SB (bSB)
machine size (# of spins)	128
computational precision	32-bit floating point
Resource (core+pre.+post.)	
ALM [count]	84,917
BRAM [count]	831
BRAM [bits]	1,442,928
DSP [count]	932
Implementation	
F_{sys} [MHz]	208
Cycles per SB step [cycle]	110
Cycles for pre. [cycle]	129
Cycles for post. [cycle]	648
Performance ($N_{\text{step}} = 300$)	
$t_{\text{pre+core+post}}$ [μs]	162.1
t_{core} [μs]	158.4
t_{pre} [μs]	0.6
t_{post} [μs]	3.1



(a)

(b)

FIGURE 3. System implementation. (a) Architecture and implementation details of the SBM module. (b) Placement of system modules in the FPGA.

circuit synthesis, placement, and routing is 208 MHz. The clock cycles of the SB main (core) processing, preprocessing, and postprocessing are 33,000 per run (110 per SB step), 129, and 648, respectively. The computation time (the module latency) per run ($t_{\text{pre}} + t_{\text{core}} + t_{\text{post}}$) is 162.1 μs , where the SBM core processing is dominant ($t_{\text{core}} = 158.4 \mu\text{s}$). The system-wide latency from the market feed arrival to order packet issuance depicted in Fig. 1(b) as a red arrow is 164 μs (including the latencies of the RX, price buffer, judgment, SBM, 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 [15]. Here we show the capability of the SBM module to find the trading opportunities of the proposed strategy using the real market data of TSE in comparison with that of a simulated annealing (SA) implementation [64].

Assuming the same $N = 128$ universe as in Sec. V, we sampled eight market situations that include at least one trading opportunity each in the period from Feb. 7, 2023 to Mar. 10, 2023 (see Fig. 4) and, for each market situation, examined all the solutions (the bit configurations $\{b_i\}$) satisfying the constraints for N_s -stock selection ($N_s = 4$ as in Sec. V) and delta-neutral positions (corresponding to Eq. 5) by a brute force method for verification. We then evaluated the cost function described in Eq. 3 for all the constrain-satisfying solutions and found the best (minimum) and worst (maximum) cost values. We executed the SBM and the SA (one time execution) for each market situation and then evaluated the cost values of the solutions obtained by the SBM and the SA after confirming that those solutions satisfied the constraints. The cost values were normalized so that the best one is 1 and the worst one is 0. This normalized cost value is referred to as the relative solution accuracy.

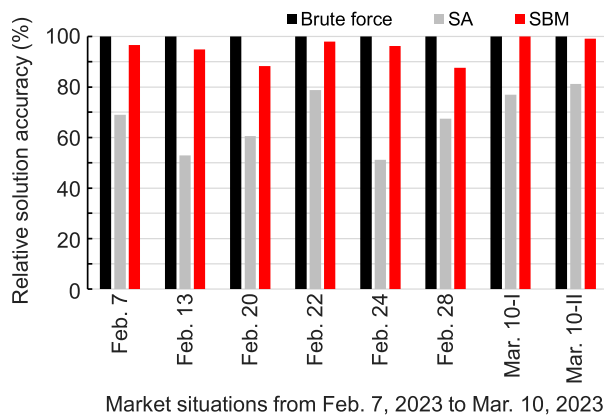


FIGURE 4. Relative solution accuracy of the SBM module and the SA implementation [64] when solving the discrete optimization problem of the proposed strategy by one execution each for the real market situations from Feb. 7, 2023 to Mar. 10, 2023.

Similarly to $N_{\text{step}} = 300$ for the SBM, the number of sweeps per execution (N_{sweep}) for the SA [40] (one sweep consists of a sequential Metropolis update of all spins) was set to be 300, so the amounts of pairwise interaction computations are similar to each other (see [16] for details). Figure 4 shows the relative solution accuracy of the SBM and the SA for each market situation. The arithmetic means over the eight market situations are, respectively, 95 % and 67 % for the SBM and the SA. Note that, for the market situation on Mar. 10, 2023, the SBM could find the exact (best) solution (the relative accuracy of 100 %) by one execution.

V. EXPERIMENT

The trading system described in Sec. IV 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 NP-hard combinatorial optimization proposed in Sec. III 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 of the trading system. The simulator assumes that the orders issued are necessarily filled at the intended prices.

Figures 5 (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 $N = 128$, $N_s = 4$, $P_{\text{max}} = 4$, and $A_{\text{trans}} = 4$ million Japanese yen (JPY). The 128 stocks were selected from the Nikkei 225 or TOPIX 100 constituents in terms of high liquidity. The system is

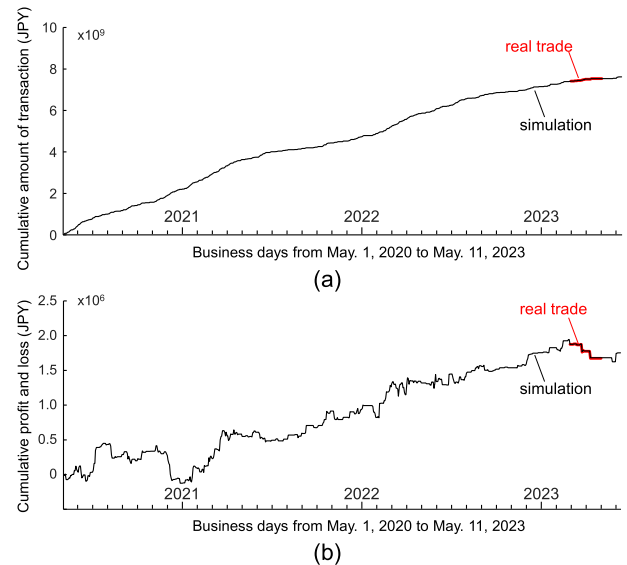


FIGURE 5. Performance of the strategy. (a) Cumulative amount of transactions in JPY and (b) cumulative profit and loss in JPY. Simulation data is from May. 1, 2020, to May. 11, 2023 (737 business days). Real trade data is from Feb. 1, 2023, to Mar. 31, 2023 (42 business days), adjusted with the simulation at the first day.

allowed to take positions in the afternoon session of the TSE. The simulation data is from May. 1, 2020, to May. 11, 2023. The real trade data is from Feb. 1, 2023, to Mar. 31, 2023, being adjusted with the simulation at the first day.

The Sharpe ratio of the strategy over the simulation period (approximately 3 years) is 1.23, where the annualized return and risk (the standard deviation of the return) are 3.6 % and 2.9 %, respectively, for an investment of 16 million JPY ($A_{\text{trans}} \times P_{\text{max}}$); the strategy proposed can be profitable (a positive annualized return) with a reasonable risk (a low level of annualized risk compared to the annualized return). The cumulative value of the amounts of transactions by the system (118,956,828 JPY) over the experiment (252 hours of real-time trading) is coincident well (+0.01 %) with the simulation value (118,948,300 JPY), indicating that the strategy proposed is executable with the trading system with a latency of 164 μs . Note that the slight difference in the transaction amounts comes from the executed prices.

Figure 6 shows a typical transaction by the trading system observed on Feb. 24, 2023. On that day, the number of the market feeds informing the changes of *ask/bid* of stocks in the $N (= 128)$ -stock universe was 5,565,723, which arrived at intervals of 3.6 ms on average. The system decided the opening of the positions at 1:13 PM in JST (15,186 seconds after 9:00 AM) based on the selection of codes 8411, 6762, 8036, and 9735 by the SBM module, leading to the profitable closing of the positions before the end of the day [Fig. 6 (a)]. The selection of the four stocks ($N_s = 4$) by the SBM module was based on the deviations of the stock prices from the VWAPs (Δp_i) and the correlation factors ($\sigma_{i,j}$) shown in Figs. 6 (b), (c), and (d). Codes 8411 and 8036 were selected mainly because of the instantaneous expected returns (the maximum and second-maximum ones at that moment) as the

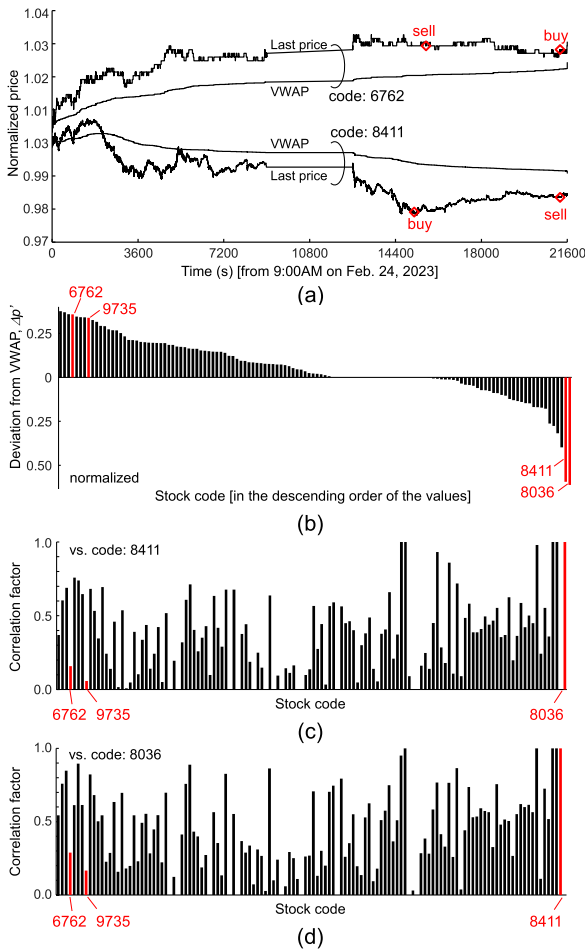


FIGURE 6. A typical transaction by the trading system observed on Feb. 24, 2023. (a) Last price vs. time on the day, where the execution times of transactions are indicated by red markers. The open decision at the time of 15,186 was made based on (b) the deviations from VWAPs of stocks, (c) the correlation factors of stocks vs. code 8411, and (d) the correlation factors of stocks vs. code 8036.

candidates for long positions. From the candidates for short positions balancing to codes 8411 and 8036, codes 6762 and 9735 were chosen based on not only the relatively high expected returns but also relatively low correlation factors against both codes 8411 and 8036. The solution of the SBM module satisfies the constraints of the discrete optimization; $\sum_i^N b_i = N_s$ and $\sum_i^N \text{sgn}(\Delta p_i) b_i = 0$ in the representation using bit variables b_i .

VI. CONCLUSION

We proposed a strategy based on selections of potentially profitable, uncorrelated, and balanced stocks by NP-hard, quadratic and discrete optimization 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 selection problem.

The cost function of N_s -stock selection problem is designed to involve maximizing instantaneous expected returns defined as deviations from volume-weighted average

prices (VWAPs), minimizing the summation of statistical correlation factors (for correlation diversification), and penalty functions for N_s -stock selection and delta-neutral positions. The selection problem is formulated in the form of the Ising problem and then the data describing the problem is separated into two components that change tick-by-tick (N -size) or day-by-day ($N \times N$ -size). By customizing the SBM core circuit to have two sets of memory and computation modules respectively for the tick-by-tick and day-by-day change problem-components, we reduced the data size transferred to the SBM core from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$ when the market situation changes and improved the system latency. This is a technique to improve the system latency when the problem components change at different timing and is applicable to the SB algorithm and other algorithms based on Hamiltonian equations of motion.

The automated trading system is a hybrid FPGA/CPU system, featuring an event-driven SBM module in the FPGA part. The FPGA part (hardware processing) decides the opening of a group of long/short positions using the SBM and then issues the corresponding orders, while the CPU part (software processing) manages the positions (including the decision of closing positions). The system-wide latency from the market feed arrival to the order packet issuance is $164 \mu\text{s}$ for a 128-stock universe. In comparison with a brute force method and a simulated annealing implementation (SA), it has been shown that the SBM module, by one execution, finds a near-optimal solution (whose accuracy is better than the SA) of the discrete optimization problem.

The trading system was installed at the JPX Co-location area of the TSE and operated for a real-time trading period of 42 business days or 252 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 trading strategy based on the NP-hard discrete portfolio optimization.

Automated trading systems with embedded Ising machines would be applicable to the strategies based on various discrete portfolio optimizations characterized by different definitions of expected returns and correlations [diagonal and non-diagonal terms in Eq. (3)] and other trading strategies that rely on high-speed discrete optimization.

APPENDIX A QUBO & ISING REPRESENTATIONS

The QUBO formulation ($b_i \in \{0, 1\}$),

$$H_{\text{QUBO}} = \sum_i^N \sum_j^N Q_{i,j} b_i b_j, \quad (7)$$

is represented also in the Ising formulation ($s_i \in \{-1, 1\}$) as follows.

$$H_{\text{Ising}} = -\frac{1}{2} \sum_i^N \sum_j^N J_{i,j} s_i s_j + \sum_i^N h_i s_i, \quad (8)$$

where

$$s_i = 2b_i - 1, \quad (9)$$

$$J_{i,j} = \begin{cases} -\frac{Q_{i,j}}{2} & (\text{if } i \neq j), \\ 0 & (\text{if } i = j), \end{cases} \quad (10)$$

$$h_i = \sum_j^N \frac{Q_{i,j}}{2}. \quad (11)$$

APPENDIX B ADDITIONAL COMPUTATION UNITS

The correction (Δy_i^{tick}) of the momentum per SB time-evolution step for i th oscillator depending on the tick-by-tick J and h components is expressed by

$$\Delta y_i^{\text{tick}} = \frac{c_3}{2} \left(\Delta Y^{\text{tick}} - x_i \right) \text{sgn}(\Delta p_i) - h_i^{\text{tick}}, \quad (12)$$

where

$$\Delta Y^{\text{tick}} = \sum_i^N \text{sgn}(\Delta p_i) x_i, \quad (13)$$

$$h_i^{\text{tick}} = -c_1 \frac{|\Delta p_i|}{2} + c_3 \left(\sum_j^N \frac{\text{sgn}(\Delta p_j)}{2} \right) \text{sgn}(\Delta p_i). \quad (14)$$

The JX^{tick} in the MAC^{tick} module (Fig. 2) is provided with the x_i and $\text{sgn}(\Delta p_i)$ data from the global X'_{mem} memory and the J^{tick} memory and calculates the ΔY^{tick} in a spatially parallel manner using multiple MAC processing elements. The time evolution (TE) module receives the ΔY^{tick} and $\text{sgn}(\Delta p_i)$ data from the MAC^{tick} module and also receives the h_i^{tick} data from the H^{tick} memory, and then updates the momentum of each oscillator by respective correction of Δy_i^{tick} in a temporal parallel manner (pipelining).

APPENDIX C 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 IV 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 \times 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

Kosuke Tatsumura, Ryo Hidaka, and Masaya Yamasaki 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.

AUTHOR CONTRIBUTIONS

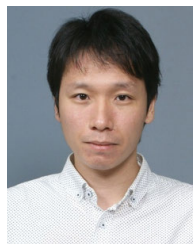
All the authors contributed to the whole aspects of this work, with each making the following major contributions. Kosuke Tatsumura conceived and managed the project, formulated the problem, and wrote the manuscript. Ryo Hidaka improved the strategy and simulator, and customized and implemented the SBM circuit. Jun Nakayama proposed the strategy and developed the simulator. Tomoya Kashimata designed the network configuration of the system and developed the interfaces of the FPGA to the extra network and to the CPU. Masaya Yamasaki architected, integrated, tested, and operated the FPGA/CPU systems.

REFERENCES

- [1] D. Bienstock, "Computational study of a family of mixed-integer quadratic programming problems," *Math. Program.*, vol. 74, no. 2, pp. 121–140, Aug. 1996, doi: [10.1007/BF02592208](https://doi.org/10.1007/BF02592208).
- [2] R. Mansini and M. G. Speranza, "Heuristic algorithms for the portfolio selection problem with minimum transaction lots," *Eur. J. Oper. Res.*, vol. 114, pp. 219–233, Apr. 1999, doi: [10.1016/S0377-2217\(98\)00252-5](https://doi.org/10.1016/S0377-2217(98)00252-5).
- [3] H. Markowitz, "Portfolio selection," *J. Finance*, vol. 7, no. 1, pp. 77–91, 1952, doi: [10.2307/2975974](https://doi.org/10.2307/2975974).
- [4] D. Venturelli and A. Kondratyev, "Reverse quantum annealing approach to portfolio optimization problems," *Quantum Mach. Intell.*, vol. 1, nos. 1–2, pp. 17–30, May 2019, doi: [10.1007/s42484-019-00001-w](https://doi.org/10.1007/s42484-019-00001-w).
- [5] J. Lang, S. Zielinski, and S. Feld, "Strategic portfolio optimization using simulated, digital, and quantum annealing," *Appl. Sci.*, vol. 12, no. 23, p. 12288, Dec. 2022, doi: [10.3390/app122312288](https://doi.org/10.3390/app122312288).
- [6] G. Rosenberg, P. Haghnegahdar, P. Goddard, P. Carr, K. Wu, and M. L. de Prado, "Solving the optimal trading trajectory problem using a quantum annealer," in *Proc. 8th Workshop High Perform. Comput. Finance*, Nov. 2015, pp. 1–7, doi: [10.1145/2830556.2830563](https://doi.org/10.1145/2830556.2830563).
- [7] K. Steinhauer, T. Fukadai, and S. Yoshida, "Solving the optimal trading trajectory problem using simulated bifurcation," 2020, *arXiv:2009.08412*, doi: [10.48550/arXiv.2009.08412](https://doi.org/10.48550/arXiv.2009.08412).
- [8] S. Mugel, C. Kuchkovsky, E. Sánchez, S. Fernández-Lorenzo, J. Luis-Hita, E. Lizaso, and R. Orús, "Dynamic portfolio optimization with real datasets using quantum processors and quantum-inspired tensor networks," *Phys. Rev. Res.*, vol. 4, no. 1, Jan. 2022, Art. no. 013006, doi: [10.1103/PhysRevResearch.4.013006](https://doi.org/10.1103/PhysRevResearch.4.013006).
- [9] S. Butenko, "Maximum independent set and related problems, with applications," Ph.D. dissertation, Dept. Ind. Syst. Eng., Univ. Florida, Gainesville, FL, USA, 2003. [Online]. Available: https://ufdcimages.uflib.ufl.edu/UF/E0/00/10/11/00001/butenko_s.pdf

- [10] 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](https://doi.org/10.1142/9789812562586_0001).
- [11] 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](https://doi.org/10.1002/9781118593486.ch4).
- [12] Y. Sakurai, Y. Yuki, R. Katsuki, T. Yazane, and F. Ishizaki, "Correlation diversified passive portfolio strategy based on permutation of assets," *J. Investment Strategies*, vol. 10, pp. 1–22, Jun. 2021, doi: [10.21314/JOIS.2021.010](https://doi.org/10.21314/JOIS.2021.010).
- [13] 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](https://doi.org/10.1126/sciadv.aav2372).
- [14] 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](https://doi.org/10.1109/FPL.2019.00019).
- [15] 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](https://doi.org/10.1126/sciadv.abe7953).
- [16] 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](https://doi.org/10.1038/s41928-021-00546-4).
- [17] 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](https://doi.org/10.35848/1882-0786/acaba9).
- [18] M. W. Johnson et al., "Quantum annealing with manufactured spins," *Nature*, vol. 473, no. 7346, pp. 194–198, May 2011, doi: [10.1038/nature10012](https://doi.org/10.1038/nature10012).
- [19] 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](https://doi.org/10.1038/s41586-023-05867-2).
- [20] 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](https://doi.org/10.1126/sciadv.abh0952).
- [21] D. Pierangeli, G. Marucci, 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](https://doi.org/10.1103/PhysRevLett.122.213902).
- [22] 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](https://doi.org/10.1038/s41928-020-0436-6).
- [23] 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](https://doi.org/10.1038/s41928-022-00774-2).
- [24] 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](https://doi.org/10.1038/s41928-022-00749-3).
- [25] 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](https://doi.org/10.1145/3470496.3527414).
- [26] T. Takemoto, M. Hayashi, C. Yoshimura, 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](https://doi.org/10.1109/JSSC.2019.2949230).
- [27] 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. 42–43, doi: [10.1109/ISSCC42615.2023.10067504](https://doi.org/10.1109/ISSCC42615.2023.10067504).
- [28] 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](https://doi.org/10.1109/ASP-DAC47756.2020.9045100).
- [29] 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](https://doi.org/10.1109/TETC.2019.2957177).
- [30] T. Okuyama, T. Sonobe, K. Kawarabayashi, and M. Yamaoka, "Binary optimization by momentum annealing," *Phys. Rev. E*, vol. 100, no. 1, Jul. 2019, Art. no. 012111, doi: [10.1103/PhysRevE.100.012111](https://doi.org/10.1103/PhysRevE.100.012111).
- [31] S. G. Brush, "History of the Lenz–Ising model," *Rev. Mod. Phys.*, vol. 39, no. 4, pp. 883–893, Oct. 1967, doi: [10.1103/RevModPhys.39.883](https://doi.org/10.1103/RevModPhys.39.883).
- [32] P. L. Krapivsky and S. Redner, "Dynamics of majority rule in two-state interacting spin systems," *Phys. Rev. Lett.*, vol. 90, no. 23, Jun. 2003, Art. no. 238701, doi: [10.1103/PhysRevLett.90.238701](https://doi.org/10.1103/PhysRevLett.90.238701).
- [33] R. Morris, "Bootstrap percolation, and other automata," *Eur. J. Combinatorics*, vol. 66, pp. 250–263, Dec. 2017, doi: [10.1016/j.ejc.2017.06.024](https://doi.org/10.1016/j.ejc.2017.06.024).
- [34] Y. Shang, "A note on the majority dynamics in inhomogeneous random graphs," *Results Math.*, vol. 76, no. 3, p. 119, Aug. 2021, doi: [10.1007/s00025-021-01436-z](https://doi.org/10.1007/s00025-021-01436-z).
- [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](https://doi.org/10.1088/0305-4470/15/10/028).
- [36] A. Lucas, "Ising formulations of many NP problems," *Frontiers Phys.*, vol. 2, pp. 5:1–5:15, Feb. 2014, doi: [10.3389/fphy.2014.00005](https://doi.org/10.3389/fphy.2014.00005).
- [37] 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](https://doi.org/10.1109/ISCAS45731.2020.9181114).
- [38] K. Tatsumura, R. Hidaka, J. Nakayama, T. Kashimata, and M. Yamasaki, "Pairs-trading system using quantum-inspired combinatorial optimization accelerator for optimal path search in market graphs," *IEEE Access*, vol. 11, pp. 104406–104416, 2023, doi: [10.1109/ACCESS.2023.3316727](https://doi.org/10.1109/ACCESS.2023.3316727).
- [39] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983, doi: [10.1126/science.220.4598.671](https://doi.org/10.1126/science.220.4598.671).
- [40] S. V. Isakov, I. N. Zintchenko, T. F. Rønnow, and M. Troyer, "Optimised simulated annealing for Ising spin glasses," *Comput. Phys. Commun.*, vol. 192, pp. 265–271, Jul. 2015, doi: [10.1016/j.cpc.2015.02.015](https://doi.org/10.1016/j.cpc.2015.02.015).
- [41] M. J. A. Schuetz, J. K. Brubaker, and H. G. Katzgraber, "Combinatorial optimization with physics-inspired graph neural networks," *Nature Mach. Intell.*, vol. 4, no. 4, pp. 367–377, Apr. 2022, doi: [10.1038/s42256-022-00468-6](https://doi.org/10.1038/s42256-022-00468-6).
- [42] 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](https://doi.org/10.1137/21M1414024).
- [43] Y. Shang, "Non-linear consensus dynamics on temporal hypergraphs with random noisy higher-order interactions," *J. Complex Netw.*, vol. 11, no. 2, pp. 509–512, Apr. 2023, doi: [10.1093/comnet/cnad009](https://doi.org/10.1093/comnet/cnad009).
- [44] 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](https://doi.org/10.1016/j.physrep.2020.05.004).
- [45] 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. Anal.*, vol. 14, no. 1, pp. 1–15, Jun. 2022, doi: [10.1007/s41060-021-00306-9](https://doi.org/10.1007/s41060-021-00306-9).
- [46] 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](https://doi.org/10.3390/e22070773).
- [47] A. O. Bielinskyi, V. N. Soloviev, S. V. Hushko, A. E. Kiv, and A. V. Matviychuk, "High-order networks and stock market crashes," in *Proc. Int. Conf. Monit., Modeling Manage. Emergent Economy (M3E2)*, 2023, pp. 134–144, doi: [10.5220/0011931900003432](https://doi.org/10.5220/0011931900003432).
- [48] S. Yoo, H. Kim, J. Kim, S. Park, J.-Y. Kim, 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](https://doi.org/10.1109/HPCA56546.2023.10070930).

- [49] M. Fil and L. Kristoufek, "Pairs trading in cryptocurrency markets," *IEEE Access*, vol. 8, pp. 172644–172651, 2020, doi: [10.1109/ACCESS.2020.3024619](https://doi.org/10.1109/ACCESS.2020.3024619).
- [50] 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](https://doi.org/10.1080/17517575.2018.1493145).
- [51] 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. E98-D, no. 2, pp. 288–297, 2015, doi: [10.1587/transinf.2014RCP0011](https://doi.org/10.1587/transinf.2014RCP0011).
- [52] 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](https://doi.org/10.1109/FPL.2011.64).
- [53] V. Betz, J. Rose, and A. Marquardt, *Architecture and CAD for Deep-Submicron FPGAs*. New York, NY, USA: Springer, 1999, doi: [10.1007/978-1-4615-5145-4](https://doi.org/10.1007/978-1-4615-5145-4).
- [54] 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](https://doi.org/10.1038/srep21686).
- [55] L. Malcencie, K. Malceniaks, and T. J. Putniņš, "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](https://doi.org/10.1016/j.jfineco.2018.02.015).
- [56] 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](https://doi.org/10.1093/rfs/hhu032).
- [57] W. F. Sharpe, "Mutual fund performance," *J. Bus.*, vol. 39, no. 1, pp. 119–138, 1966. [Online]. Available: <https://www.jstor.org/stable/2351741>
- [58] 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](https://doi.org/10.1111/j.1540-6261.1993.tb05132.x).
- [59] 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](https://doi.org/10.1093/rfs/hhj020).
- [60] S. A. Berkowitz, D. E. Logue, and E. A. Noser Jr., "The total cost of transactions on the NYSE," *J. Finance*, vol. 43, no. 1, pp. 97–112, Mar. 1988, doi: [10.1111/j.1540-6261.1988.tb02591.x](https://doi.org/10.1111/j.1540-6261.1988.tb02591.x).
- [61] S. M. Kakade, M. Kearns, Y. Mansour, and L. E. Ortiz, "Competitive algorithms for VWAP and limit order trading," in *Proc. 5th ACM Conf. Electron. Commerce*, May 2004, pp. 189–198, doi: [10.1145/988772.988801](https://doi.org/10.1145/988772.988801).
- [62] J. Białkowski, S. Darolles, and G. Le Fol, "Improving VWAP strategies: A dynamic volume approach," *J. Banking Finance*, vol. 32, no. 9, pp. 1709–1722, Sep. 2008, doi: [10.1016/j.jbankfin.2007.09.023](https://doi.org/10.1016/j.jbankfin.2007.09.023).
- [63] G. Marsaglia, "Xorshift RNGs," *J. Stat. Softw.*, vol. 8, no. 14, pp. 1–6, 2003, doi: [10.18637/jss.v008.i14](https://doi.org/10.18637/jss.v008.i14).
- [64] D-Wave Systems. *DWave-Neal (Version: 0.5.9)*. Accessed: Feb. 9, 2022. [Online]. Available: <https://github.com/dwavesystems/dwave-neal>



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. In 2008, he joined Toshiba Corporation. 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™, and host controllers for flash-memory cards. His current research interests include domain-specific computing, high-level synthesis design methodology, and proof-of-concept study with FPGA devices.



JUN NAKAYAMA received the B.A. degree in economic and social studies from The University of Manchester, U.K., in 2008, and the M.B.A. degree in finance from Hitotsubashi University, Japan, in 2017. From 2008 to 2020, he was the Portfolio Manager of Nomura Asset Management Company Ltd. He was engaged in the development and management of quant-based funds. In 2020, he joined the Corporate Research and Development Center, Toshiba Corporation. 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. In 2020, he joined the Corporate Research and Development Center, Toshiba Corporation, Japan. His research interests include computer architecture, reconfigurable architecture, and processing-in-memory (PIM).



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 a Chief Research Scientist, leading a research team and several projects toward realizing innovative industrial systems based on cutting-edge computing technology. From 2013 to 2015, he was a member of the Emerging Research Devices (ERD) Committee in the International Technology Roadmap for Semiconductors (ITRS). From 2015 to 2016, he was a Visiting Researcher with the University of Toronto. Since 2013, he has been a Lecturer with Waseda University. His research interests include domain-specific computing, quantum/quantum-inspired computing, and their applications. He received the Best Paper Award from the IEEE International Conference on Field-Programmable Technology (FTP), in 2016.



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. In 1999, he joined Toshiba Corporation. 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 proof-of-concept study with FPGA devices.

...