Hindawi
Scientific Programming
Volume 2018, Article ID 8721246, 12 pages
https://doi.org/10.1155/2018/8721246

Research Article

Automatic High-Frequency Trading: An Application to Emerging Chilean Stock Market

Broderick Crawford,[1] **Ricardo Soto**,[1] **Marco Alarcón San Martín**,[1] **Hanns de la Fuente-Mella**,[2] **Carlos Castro**,[3] **and Fernando Paredes**[4]

[1]*Pontificia Universidad Católica de Valparaíso Chile, Avenida Brasil 2241, Valparaíso 2362807, Chile*
[2]*Pontificia Universidad Católica de Valparaíso Chile, Avenida Brasil 2830, Valparaíso 2340031, Chile*
[3]*Universidad Técnica Federico Santa María Chile, Avenida España 1680, Valparaíso 2390123, Chile*
[4]*Universidad Diego Portales Chile, Av. Ejército 441, Santiago 8370109, Chile*

Correspondence should be addressed to Hanns de la Fuente-Mella; hanns.delafuente@pucv.cl

Received 8 March 2018; Revised 11 August 2018; Accepted 5 September 2018; Published 30 September 2018

Academic Editor: José E. Labra

This research seeks to design, implement, and test a fully automatic high-frequency trading system that operates on the Chilean stock market, so that it is able to generate positive net returns over time. A system that implements high-frequency trading (HFT) is presented through advanced computer tools as an NP-Complete type problem in which it is necessary to optimize the profitability of stock purchase and sale operations. The research performs individual tests of the algorithms implemented, reviewing the theoretical net return (profitability) that can be applied on the last day, month, and semester of real market data. Finally, the research determines which of the variants of the implemented system performs best, using the net returns as a basis for comparison. The use of particle swarm optimization as an optimization algorithm is shown to be an effective solution since it is able to optimize a set of disparate variables but is bounded to a specific domain, resulting in substantial improvement in the final solution.

1. Introduction

This research seeks to design, implement, and test a fully automatic trading system that operates on the national Chilean stock market, so that it is capable of generating positive net returns over time. In particular, it is desired to create a system that implements high-frequency trading (HFT). Thus, the research corresponds to the application of advanced computer tools to a problem of type NP-Complete, where the aim is to optimize the profitability of operations of purchase and sale of shares. In this way, the objective is to create an implementation of an automatic trading system that is capable of generating positive returns for a set of real data of the national stock market, under a completely automatic modality, where there is no intervention of a human operator in the decision-making and execution of operations. Section 2 describes automatic and semiautomatic stock-trading systems and algorithmic high-frequency trading context. Section 3 provides a review of current trading algorithm techniques that can work under the automatic mode of HFT, indicating which computer techniques can be applied. Section 4 presents the design of an automatic trading system, in HFT mode, indicating the restrictions on the data and financial instruments included in the study. Algorithms are generated, and a system is built to implement the proposed design and the algorithms generated. Individual tests of the implemented algorithms are carried out, reviewing the theoretical net return (profitability) that can be generated applied on the last day, month, and semester of real market data. Finally, in Section 5, it is determined which of the variants of the implemented system behaves better, using the net returns as a basis for comparison and applying other criteria as deemed necessary.

2. Background

Stock trading is an activity that has been conducted for hundreds of years and is currently performed on stock exchanges around the world. In these exchanges, a huge variety of financial assets and debt instruments are traded daily. Stock trading is a complex decision-making problem that involves multiple variables and does not always have an optimal solution, since the conditions vary over time and are affected by internal and external factors.

In recent years, implementation of automatic and semi-automatic stock-trading systems that can analyze market conditions and make the necessary decisions to conduct required business transactions have begun. Since 2008, such systems have reported significant and constant gains on foreign stock exchanges such as the New York Stock Exchange (NYSE) [1]. These systems can be traced back to Morgan Stanley's "Black Box" system, which was created in 1985 [2] for the generation of buy/sell signals based on statistical arbitrage.

The benefit of using automated systems for the stock trading task lies in the fact that people tend to pay too much attention to signals (price) and ignore the transition probabilities that generate the signals. An automated system, in contrast, can calculate the probabilities of price transition and act accordingly [3], avoiding problems of late reaction or overreaction to changes.

At present, the entry of automatic traders into the Chilean market has been relatively low, and there is no information available on such entry since those who employ automatic trading mechanisms are unwilling to divulge the details of their systems for fear of competition, as occurred in the beginning with the "Black Box" system. The domestic market has been able to operate with automatic low- and high-frequency traders since 2010, when the Santiago Stock Exchange launched the Telepregon HT system, which allows the trading of equities at a theoretical maximum rate of 3000 transactions per second [4, 5].

Trading is the exchange of ownership of a good, product, or service from a person or entity under conditions in which something is obtained in return from the buyer. Thus, trading can be understood as the practice conducted by stockbrokers or their clients whereby financial instruments are exchanged in securities markets. This trading is based on the principle of supply and demand of the traded instruments, which causes the prices of the instruments to vary and generates a profit (or loss) that is determined by the difference between the original purchase price and the final sale price.

High-frequency trading (HFT) is understood as a way of operating in stock markets to which a number of special conditions [1] apply:

(i) There is a rapid exchange of capital

(ii) A large number of transactions are performed

(iii) Generally, a low gain per transaction is obtained

(iv) Financial instrument positions are neither accumulated from one trading day to another nor avoided

(v) Trading is conducted through a computer system

The definition of HFT itself does not indicate whether the system performing it is automatic, semiautomatic or user-operated.

In contrast, automatic trading varies from systems that support the entry of buy/sell orders to the market, such as systems capable of entering orders automatically without the need for a human operator but that can maintain the positions of financial instruments from one day to the next.

There is no single formula for defining an HFT or an automatic trading system [1, 6, 7]. For example, it is stated that an algorithmic trading (AT) system corresponds to "elements in decision-making and financial investments being executed by an algorithm through computers and electronic communication networks" [7]. Investment strategies can be predefined or adaptive. These investment strategies can be supported by knowledge of economics, statistics, artificial intelligence, metaheuristics, etc.

Similarly, it is proposed a sequential process for developing an HFT system that is based on four steps: (i) data analysis; (ii) trading model; (iii) decision-making; and (iv) execution of business [7].

Thus, there is no single formula for producing an HFT system. However, it is worth noting that to achieve an effective HFT system, it is necessary to take into account a series of processes common to any system, namely, analysis, identification, collation, routing, and execution [8].

In any of the automated systems described above, the components that present the greatest complexity are the analysis of real-time opportunities and the search for market inefficiencies. The available literature mentions methods of the following types: (i) Rule-based methods such as statistical arbitration [2]. These methods apply a series of rules that are based on the recent behavior of a financial instrument and act based on the result of applying those rules. (ii) Methods based on statistical and mathematical models, such as volume-weighted average price, time-weighted average price, and moving averages [1, 7]. For these cases, a mathematical or statistical model is used that requires a series of parameters that control its behavior. The selection of the configuration parameters is performed by a manual operator who is in charge of trading on the market. (iii) Methods combining statistical and mathematical models with optimization techniques based on metaheuristics [9]. These methods use metaheuristics to automatically fine-tune the parameters of known algorithms to obtain optimum values for current market conditions. (iv) Methods based on machine learning, data mining, and processing of complex events [1, 9, 10]. Because rapid assimilation of the large amount of information flowing to and from any stock market by a human operator is becoming an increasingly difficult task; it is desirable to develop systems that are able to detect hidden patterns in price variations and the relationships between financial instruments or other economic indicators and that can also incorporate a component of interpretation of "feeling" or "sensing" the market through natural language news processing (e.g., SuperX Plus of Deutsche Bank [11]).

3. Trading Algorithms

3.1. Statistical Methods Used in AT and HFT. Some of the most popular trading algorithms based on statistical or mathematical methods [7, 12] are as follows:

Volume-weighted average price (VWAP) is defined as the ratio of the volume of transactions rated against the volume of the instrument over the trading horizon. It is common to evaluate the performance of traders by their ability to execute buy/sell orders at prices that are better than the VWAP price on the trading horizon. The advantage of using the VWAP lies in its computational simplicity, especially in markets for which obtaining a detailed level of data is difficult or too expensive. The VWAP for an instrument i on a day T is calculated as follows:

$$\mathrm{VWAP}_i = \frac{\sum_t v_{\mathrm{it}} p_{\mathrm{it}}}{\sum_t v_{\mathrm{it}}}, \quad (t \in T), \tag{1}$$

where v_{it} is the volume of the instrument i traded at time t, and p_{it} is the market price of instrument i at time t. VWAP can be used to minimize the costs of transactions and market impacts. It can also be used as a benchmark to verify the effectiveness of other algorithms and trading strategies.

A modification to the model called DVWAP (dynamic VWAP) is proposed by [13]. This modification allows intraday transactions (transactions realized during the same day of execution) to be incorporated. This allows the model to be applied to a more realistic scenario of the market in which the news that arrives affects the price of the instruments.

Time-weighted average price (TWAP) is the average price of a financial instrument over a specific period of time during which the order is executed at the TWAP price or better. It is used to execute orders at a specific time to keep the price close to what the market reflects at that time. The TWAP of an instrument i in a period T is calculated as follows:

$$\mathrm{TWAP}_i = \frac{1}{T} \sum_{t=1}^{T} p_{\mathrm{it}}, \quad (t \in T), \tag{2}$$

where p_{it} is the market price of the instrument i at time t. Like VWAP, TWAP can be used as a benchmark to verify the effectiveness of other algorithms and models.

Other types of algorithms include variants of the linear econometric models presented by [1]. These models attempt to predict the behavior of random variables as a combination of other random variables, both contemporaneous and retrospective, with well-defined distributions. Such linear models can be expressed as

$$y_t = \alpha + \sum_{i=0}^{\infty} \beta_i x_{t-i} + \sum_{j=0}^{\infty} \gamma_j z_{t-j} + \ldots + \varepsilon_t, \tag{3}$$

where $\{y_t\}$ is the time series of a random variable on which a forecast is to be made; $\{x_t\}$ and $\{z_t\}$ are significant factors for predicting the value of $\{y_t\}$; α, β, and γ are the factors to be determined; and εt is the remaining error.

Moving averages (MA), a model for predicting future movements in the price of a financial instrument, focuses on how future data will react to changes in past data. To generate the MA model (q) with q delays, we use

$$y_t = c_0 + a_t - \theta_1 a_{t-1} - \ldots - \theta_q a_{t-q}, \tag{4}$$

where c_0 is the intercept, θ_l is the coefficient belonging to delay l, and al is the unexpected component of the return at delay l. There are several ways to estimate MA; they include the following:

Simple MA (SMA) is the weightless average of the previous prices n. This can be n of previous days or another measure of time. It can also be calculated based on the SMA of the previous period, simplifying its calculation at the computational level.

$$\begin{aligned} \mathrm{SMA}_t &= \frac{\sum_{i=0}^{n} p_i}{t}, \\ \mathrm{SMA}_t &= \mathrm{SMA}_{t-1} - \frac{p_{t-n}}{n} + \frac{p_t}{n}, \end{aligned} \tag{5}$$

Cumulative MA (CMA) is a moving average in which all prices are considered until the current instant. Its formula is similar to that of SMA but begins from the first recorded market price for an instrument. It has no known application in trading strategies.

Weighted MA (WMA) is an average that uses multiplication factors to give different weights at different prices within the same MA window (convolution of data points with a fixed weight function). In trading, decreasing weight is assigned from n to 1 at each price in the evaluation window, as follows:

$$\begin{aligned} \mathrm{WMA}_t &= \frac{\sum_{i=0}^{n} (n-i) p_i}{\sum_{i=0}^{n} n - i}, \\ \mathrm{WMA}_t &= \frac{\sum_{i=0}^{n} (n-i) p_i}{(n(n+1)/2)}, \end{aligned} \tag{6}$$

Like MA, WMA provides a smoothing function of the prediction curve. In some cases, it is used together with MA; it can also be used when the prices of previous days do not greatly affect the value of the current price of an instrument.

Exponential MA (EMA), also referred to as EWMA (exponential weighted MA), is a version similar to WMA in which the weight is an exponential rather than a linear function. The weight assigned to each market price decreases exponentially and never reaches zero. Thus, for a series y, the EMA is calculated recursively as

$$\begin{aligned} y_0 &= \beta p_0, \\ \text{for } t > 0, \; y_t &= \alpha p_t + (1-\alpha) y_{t-1}, \end{aligned} \tag{7}$$

where α is the coefficient of decreasing weight (a constant value between 0 and 1). A high coefficient value causes the old prices to decrease more quickly. Alternatively, α can be expressed in terms of n periods of time:

$$\alpha = \frac{2}{n+1}. \tag{8}$$

3.2. Metaheuristic Models. Several known trading models and algorithms have been described in the literature. Such algorithms are generally applied manually by a human operator to determine when to buy, sell, or maintain the current position. Robert Pardo states that for a given combination of strategies, it is possible to apply optimization to determine a set of parameters that generates greater gains [9].

Such a postulate does not come without associated problems. The major known problem is that such optimizations can cause overperformance of the algorithm with respect to the data used. In the best-case scenario, the resulting algorithm will not generate the expected gains, and in the worst case, the algorithm will produce constant losses. One way to understand the concept of overperformance is to think of a statistical model that describes random error or noise instead of describing relationships between variables.

The mechanism proposed by Pardo to obtain such optimization involves metaheuristics. In this respect, both AT and HFT can be understood as complex optimization problems. This class of problems is referred to as class NP (nondeterministic polynomial time). The NP class is the class of problems in which a solution can be verified by a polynomial time algorithm but in which given the difficulty of the problems, there is no algorithm that can generate solutions in polynomial time. This implies that the application of conventional algorithms to this class of problems results in execution times that increase exponentially as the size of the problem increases.

Until 1971, there had been no demonstration of a problem of this kind. That year, Stephen Cook demonstrated the first NP-Complete practical problem [14]. In 1972, Richard Karp expanded on Cook's idea, demonstrating a series of 21 NP-Complete class problems [15].

Because AT and HFT are both problems of trading financial instruments in markets with varying conditions over time, they can both be categorized as NP-class problems [16]. Because both are based on the maximization of net returns, according to Chang and Johnson [17], they can be classified as NP-Complete, even in versions that perform offline market simulations.

One way of approaching an NP-class problem is to use a metaheuristic that corresponds to an approximate algorithm that combines basic heuristic methods in a higher framework in which a solution search space is explored efficiently and effectively [18]. Thus, through an objective function that guides the search process, an efficient exploration of possible solutions is made in search of one or more near-optimal solutions.

A great variety of metaheuristic algorithms are available. Some of these algorithms have more affinity for certain types of problems than others, such as problems with binary, discrete, or continuous variables. Some algorithms can be applied to only one variable type, or adjustments must be made such as applying conversion functions. In particular, an approach to one of the existing algorithms called particle swarm optimization (PSO) will be presented.

3.2.1. Particle Swarm Optimization. The PSO algorithm was introduced by Kennedy and Eberhart in 1995 [19] in an attempt to describe the social behavior of flocks of birds or schools of fish and to model their communication mechanisms as a basis for solving optimization problems. Instead of relying on the "survival of the fittest," PSO is based on the cooperation of individuals.

PSO is defined as a metaheuristic algorithm that optimizes a problem by iteratively improving a population of candidate solutions called particles by moving them through the solution space using a formula based on each particle's position and velocity. The movement of each particle is influenced by its best-known local solution and is also guided to the best-known global solution. With this, the swarm is expected to move collectively toward the best solution in the search space.

In the basic version of PSO, the velocity and position of the particles are calculated as follows:

$$\begin{aligned} v_i^{k+1} &= c_0 v_i^k + c_1 r_{1,i}^k \left(x_i^{k,\text{best}} - x_i^k \right) + c_2 r_{2,i}^k \left(x_{i,\text{swarm}}^{k,\text{best}} - x_i^k \right), \\ x_i^{k+1} &= x_i^k + v_i^{k+1}, \end{aligned} \tag{9}$$

where x_i^k is the position of the i-th particle at iteration k, v_i^k is the velocity of the i-th particle at iteration k, c_0 is the inertia factor (a value between 0 and 1), c_1 is the local acceleration factor (cognitive component of the individual), c_2 is the global acceleration factor (social component of the swarm), $r_{1,i}^k$ and $r_{2,i}^k$ are random numbers with uniform distributions between 0 and 1, $x_i^{k,\text{best}}$ is the best previous position of the i-th particle, and $x_{i,\text{swarm}}^{k,\text{best}}$ is the best previous position of the neighborhood of the i-th particle.

Various formulations exist for the selection of the parameters c_0, c_1, and c_2. One of the options is to use the values suggested in [20].

Other ways of determining the parameters include functions that modify the parameters during the execution of the algorithm. An example of this is given by Fikret in [21]; the example is based on the fact that the value of the parameter of inertia, c_0, influences the diversification (exploration of the search space) and intensification (exploitation of the search space). High values of the parameter of inertia favor diversification, whereas low values favor the intensification of local solutions. In this way, an exponential function of the inertia parameter is defined by

$$c_0 = c_{0,\text{end}} + \left(c_{0,\text{start}} - c_{0,\text{end}} \right) e^{k \propto / k_{\max}}, \tag{10}$$

where $c_{0,\text{start}}$ is the initial inertia, $c_{0,\text{end}}$ is the final inertia, k represents the current iteration, $k_{\max}$ is the maximum number of iterations to be performed, and α is a gradient constant. Other variants of the c_0 calculation include linear descent of the inertia parameter or a stochastic function associated with inertia.

For the specific problem of HFT and AT, the PSO algorithm is applied to the optimization of the parameters of a trading strategy based on the MA of two or more bands. In this case, the temporal parameters of the MA involved in the strategy against an objective function are optimized, including the following:

(i) Obtain the highest net return (earnings)

(ii) Obtain the most benefit per transaction

(iii) Obtain the highest percentage of winning transactions or assure that the strategy has a higher specific financial ratio

In this way, the objective function that is applied to the PSO algorithm measures and classifies the quality of the trading strategy that is applied in the AT or HFT system. Thus, the success of applying PSO to an HFT and AT problem depends primarily on how the objective function is proposed for the same trading model.

4. Methodology

The main objective of the research is to create a system that can conduct trading autonomously. Thus, a preliminary design of a system that can be applied during a full trading day for a given stock market is defined. As an initial step, this requires defining and delimiting the target market since there are multiple stock exchanges in the world, each offering a range of different markets and possessing specific regulations and restrictions.

The data used corresponds to the transactions carried out during 2 years for one of the equity instruments listed in the old IPSA index (now replaced by S&P/CLX IPSA). This corresponds to a highly liquid stock instrument in the national market. In particular, 2 sets of data were used. One is public: the register of daily operations, which is reported to the CMF (Commission of Financial Markets) and published daily in the institutional site of the Santiago Stock Exchange. The other is a product of data that is marketed: market replay, which contains the data of offers entered into the system at the level of nanoseconds, anonymized by "Corredor de Bolsa" (does not expose sensitive information of the orders as the client to which it belongs, % amount visible, internal operator, etc.).

Once the target market, data selected, and the instruments involved have been defined, a system can be designed that is capable of operating on the defined market and adapting the regulations and restrictions that govern it. The same definition of the target market and the instruments will serve to determine what external data will be required and how these data should be collected and treated by the system.

Strategies, especially classic trading strategies based on MA, should be validated in conjunction with parameter optimization using PSO. Regardless of the strategy adopted, the system that is designed must support any type of strategy, so it must be a generic and easily extensible system.

4.1. Selection of Markets and Financial Instruments. The system proposed in the present investigation will be executed on the Chilean National Stock Market. This corresponds to the entire market of equity instruments in national currency (National Shares). For this market, the system is required have the following characteristics:

(i) It has a defined operating schedule. The National Stock Market operates from 09:30 to 17:00 in the summer and from 09:30 to 16:00 in the winter. During this time, it is possible to negotiate (enter offers and modify or cancel them). There is a time slot between 09:00 and 09:23 (plus an interval of random time between 0 and 5 minutes) called the PreOpen session during which it is possible to enter or cancel offers before they are executed (with other offers).

(ii) In the case of a brokerage firm, the costs are known (stock exchange rights), and the existing regulation is exhaustive, mainly in the guarantees that each stock broker must maintain to continue operating. In the case of a particular investor, the costs vary according to each stock brokerage, but they are also known (fixed costs and variable commissions).

(iii) It is possible to identify the stocks that are the most liquid when reviewing the composition of the IPSA (Selective Stock Price Index).

(iv) It provides both electronic trading mechanisms with support for high-frequency and electronic communication mechanisms for the entry of orders. For the latter, mechanisms are provided for institutional negotiators (brokers and financial institutions) through DMA (Direct Market Access) mechanisms on the FIX 4.4 protocol, such as retail market mechanisms (for noninstitutional users) provided by brokers, using routing command clients on FIX 4.4 with command polling.

Thus, shares of the national equity market with high presence and/or belonging to the IPSA indicators that are not suspended will be used. The purpose of this is not to discard actions that do not meet this criterion for analysis and storage purposes but simply to avoid using them during normal execution of the AT/HFT system unless they change their condition to high presence. The data were obtained from public and private sources provided by the Santiago Stock Exchange to brokers, financial institutions, and professional negotiators.

4.2. AT/HFT System Design. The system is based on five annex modules and a central module for model execution. The central module is responsible for maintaining one or more trading models through a daily review of market behavior.

The basic form of operation of the execution model module consists of a parallel copy of the trading model chosen by each valid instrument in the target market. Each copy accesses the annexed modules independently to request information and to access communication interfaces, etc., but the annexed modules handle a single copy (singleton).

The system allows parallel executions. There is one thread per instrument with the possibility of trading; each thread in the chosen model is adjusted to the needs and characteristics of the instrument.

The thread requests its configuration parameters (which the human operator can change between executions) at the start of its cycle. It then requests updated market information and uses this information to load the model. The Storage process evaluates whether it is necessary to update its information; if the information is out of date, it looks for new information both in the market and in other sources of data. In either case, the process returns updated or recent market information to the model executor. The model executor evaluates the model and verifies whether there is a favorable condition for the purchase. If such a condition exists, the thread requests a risk assessment from the Risk module. If the Risk module determines that the market condition and the risk parameters are correct, the Risk module authorizes the transaction. Then, part of the capital available to make the purchase is reserved, and this part of the capital is requested by the module that handles capital and custody. With the available capital, the parameters of the order are calculated; the Communications module then sends the purchase order to the market. When the order has been entered into the market, the available capital is updated. The process is repeated cyclically throughout the trading hours. At the end of each cycle, it is possible to apply a complete revision of the model to adapt it to the new market conditions. The process for sales is similar, but it manipulates the custody of the instruments rather than the available capital.

4.3. Adaptive Model with PSO. An adaptive AT/HFT model is proposed based on some known MA strategies. For this case, we present a classic model of two MA, one long and one short, in conjunction with two bands of risk management by stop-loss and stop-win. In this model, there are four parameters to optimize, as shown in Table 1.

Table 1 shows the variables involved in the model. These variables are subject to the following restrictions:

$$\begin{aligned} x_1 &> 1, \\ x_2 &\geq 1, \\ x_1 &> x_2, \\ x_1, x_2 &\in \mathbb{N}, \\ x_3, x_4 &\in \mathbb{R}]0,1[. \end{aligned} \tag{11}$$

The principle of a 2-MA strategy is to identify when there is a crossover, that is, when the short MA curve intersects the long MA curve.

The curves can cross from below when the short MA curve intersects the long MA curve from a lower to a higher value or from above when the short MA curve intersects the long MA curve from a higher to a lower value. When a crossover of the first type (increasing) occurs, a favorable condition for the purchase occurs, since the price tends to be high. When the crossover is of the second type (decreasing), a condition is generated that discards purchases and forces custody to be liquidated through sales.

Table 1: Variables for PSO adaptive model with 2-MA strategy with mobile bands.

Variable	Description	Type
x_1	Time for long MA	Whole
x_2	Time for short MA	Whole
x_3	Percent stop-loss band	Real]0, 1[
x_4	Percent stop-win band	Real]0, 1[

The control bands are applied to the set of two MA to generate stop-loss and stop-win mechanisms integrated in the model. These bands fulfill the objective of optimizing the model since the upper band prioritizes capital over gains (thus making the capital available to the other concurrent execution threads of the trading model), and the lower band reduces the losses.

The most important feature of the PSO model is the objective function that is used. The objective function will be performed in the first instance based on optimizing the net return of the system. In this way, the objective function will be

$$\text{maximize}: \sum_{i=0}^{n} cv_i pv_i (1 - v_i) - cc_i pc_i (1 + v_i) - f_i, \tag{12}$$

where cv_i is the quantity sold in the i-th period within the simulation horizon, pv_i is the sale price of the i-th period for the only instrument traded in the simulation, cc_i is the quantity purchased in the i-th period within the simulation horizon, pc_i is the purchase price of the i-th period for the only instrument traded in the simulation, v_i are the variable costs of the i-th period required for transacting, and f_i are the fixed costs of the i-th period required for transacting.

This objective function is the calculation of the net returns for a time span of n equal and consecutive periods. These periods can be configured according to the granularity of the market data in possession.

In the case in which the two simulations obtain the same value of the objective function, the system passes to the next exclusion criterion, in which the benefit per operation is maximized. This can be interpreted as maximizing the profit obtained between a purchase and its subsequent sale. The application of this criterion is in many cases difficult to calculate since the simulation must replace orders that participated in real order executions, which does not always make the quantities tally. In this case, it may be difficult to recreate the series of sales operations that correspond to a previous purchase. Thus, when it is a tie, it is better to move to the third criterion of discrimination, which corresponds to computing and comparing Sharpe's ratio or another valid financial ratio.

The Sharpe ratio is defined as follows [1, 9]:

$$S = \frac{E[r] - r_f}{\sigma[r]}, \tag{13}$$

where

$$E[r] = \frac{\sum_{i=1}^{T} r_i}{T},$$
$$\sigma[r] = \sqrt{\frac{\sum_{i=1}^{T} (r_i - E[r])^2}{T-1}}. \tag{14}$$

Finally, Sharpe's ratio for AT/HFT processes is

$$S_{\mathrm{HFT}} = \frac{E[r]}{\sigma[r]}, \tag{15}$$

where r_i represents the net returns of the i-th period, r_f is the risk-free benchmark rate (a constant representing the opportunity cost), and T is the number of periods.

Having calculated the Sharpe ratio, the objective function appears as

$$\text{maximize}: \sum_{i=0}^{n} cv_i pv_i (1 - v_i) - cc_i pc_i (1 + v_i) - f_i, S_{\mathrm{HFT}}. \tag{16}$$

With this objective function, the adaptive model can be generated by applying a PSO algorithm that exploits the best combination of the variables defined for the problem. This makes it possible to have a rapid and effective model that is adapted to the changing market state. In the case of a tie in the profitability of the solutions, the ratio chosen for the case can be applied.

4.4. Implementation of the System. A modular system that allows extensibility (PSO automatic trading method) was built. The PSO implementation modules and the automatic trading engine have been separated. Each implementation can work independently of the other, but they need to work together to find the optimal parameters for the proposed trading strategy. The PSO module consists of the central implementation of metaheuristics but does not include the elements of a particular problem (Figure 1).

In this way, the module consists of an optimizer that requires three interfaces for its operation (Figure 1). The 3 interfaces are responsible for the following tasks:

SwarmConfigurator: Implementations of this interface must deliver a swarm composed of particles that extend to the abstract Particle class. Thus, the swarm configurator must create the initial particle configuration for a particular problem.

ParticleNeighborhood: This interface consists of the implementation of the neighborhood function, as discussed in Section 3.2.1. It has the function of determining which is the best particle $x_{i,\mathrm{swarm}}^{k,\mathrm{best}}$ within the neighbors of a given particle so that the velocity and position calculations of the particles of the swarm can be executed.

StopCriteriaEvaluator: The optimizer requires that the stop mechanism of the algorithm be indicated. Implementations of this interface must determine at the end of each optimization cycle whether execution can continue. As a basis for determining this, it is given a series of relevant data such as the number of iterations performed and the complete state of the swarm.

To adapt it to the particular problem that is to be optimized, the optimizer requires that the process be extended to the abstract implementation of the particle. The SwarmConfigurator class is responsible for instantiating the required implementation and for the implementation of the annexed interfaces. These interfaces consist of the following:

Particle, the abstract class of a particle, contains the particle's position and velocity function as interfaces. It stores the best local solution found for velocity calculation purposes. The subclasses that extend it must implement a method that generates the value of the objective function together with the implementation of a method that can be compared against another particle by the value of its objective function to determine which has a better value. For practical purposes, the criterion is used that one particle is better than another if it has a higher value for the objective function.

Position is the interface representing the position of a particle that corresponds to one of the solutions to the problem. The classes that implement it must be able to calculate the distance to another position to create different neighborhood topologies. They must also accept an implementation of the Velocity interface and apply it to their current values, generating a new position.

Velocity is the interface that represents the velocity function of a particle. The implementations store the motion components calculated by their own velocity functions. These motion components are then applied to a Position implementation by a particle.

In the first instance, at least one implementation of the interfaces and abstract classes presented was performed to solve the automatic trading problem.

Figure 2 shows the implementations of the neighborhood interfaces and the stop criterion. The implementations of the swarm configurator, particle, velocity function, and position representation occur within the automatic trader; this is discussed in the next section. The details of each implementation are as follows:

GBestParticleNeighborhood is a global neighborhood function in which the best particle among the entire swarm set is sought. It is the simplest and fastest since it requires only finding the particle that maximizes the objective function for an iteration.

LBestParticleNeighborhood is a local neighborhood function in which the two particles closest to a given particle are searched and the best of the three particles is chosen. This is slower because it requires performing distance calculations between all the particles to find the particles that are closest to each other.

BasicStopCriteriaEvaluator is a detention criterion that is based on the number of iterations performed. When the number of designated iterations has been reached, the PSO algorithm stops. When it has stopped, the best approximate solution to the overall optimum is obtained.

4.5. Implementation of the Automatic Trader. For the implementation of the automatic trading engine, there is a central module that performs the necessary coordination

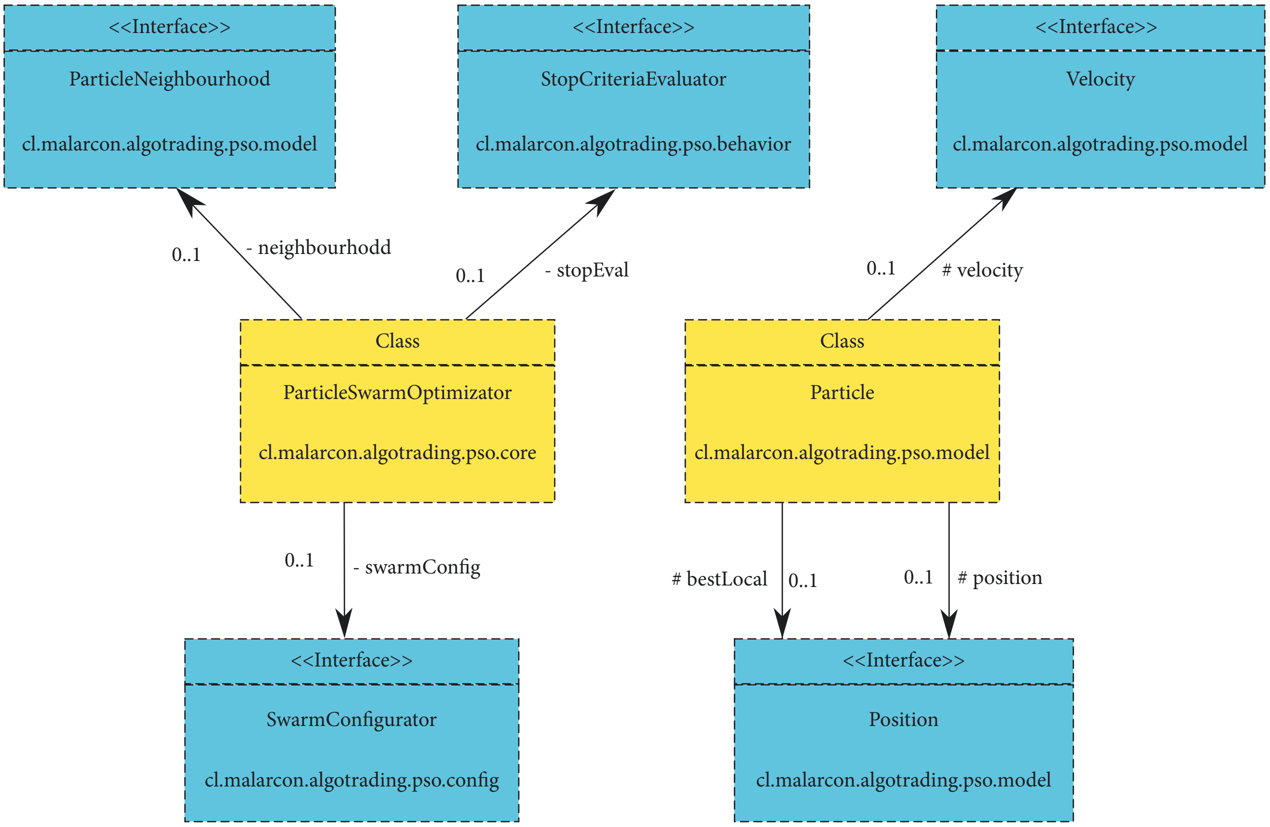


Figure 1: Basic PSO implementation model. It has a central optimizer that works with any problem that is modeled using the exposed interfaces.

to process the information related to a financial instrument through annexed modules that are specialized to perform specific tasks. The central module corresponds to an abstract class of automatic trading logic that can be generalized to any type of stock market (equities, fixed income, etc.); it uses a series of interfaces to access specialized modules in the target market. The central module operates at regular intervals (ticks), and it evaluates its internal trading algorithm during each run of the interval. If a buy/sell signal is generated, it proceeds to use the corresponding modules to perform the operation. The tick value determines whether the system behaves as an HFT system or as an AT system.

For its correct functioning, the central class of the system must be extended to generate the necessary functionality for a specific type of trader. At least two types of traders are required: one market simulation trader and one trader that communicates with the real market.

For the initial version of the AT system, the implementations of the interfaces for the simulation engine required by the PSO algorithm are created. This implementation simulates an extended period of the market through data uploaded to an MS SQL server database. In this way, the maximum profitability that generates a set of parameters of the proposed trading model for the selected period can be calculated.

The complete set of the generic PSO system and the AT engine are combined by a simple boot system to make it possible to perform the laboratory tests with the data collected in Market Maker. The boot system is configured based on a text file, and a parameter that indicates which mnemonic is to be entered into the optimizer.

Based on the laboratory tests performed, a number of improvements were made in the implementation of the system, generating an optimized version for performance. The purpose of this is to ensure that the optimization process of the solution using PSO converges rapidly enough to be executed multiple times during a day of trading.

The improvements applied address the following problems:

MA Calculation: The initial version of the AT system invokes the routine calculation of MA for each instant of system operation independently for each particle. The point calculation is changed by an incremental calculation based on the values of the previous and the new time period and on the totality of instances required by the execution of a particle.

MA Reading: Continuing from the previous problem, if for a given instant an MA with the same length had already been calculated, it was nevertheless recalculated. This introduces an overload to the Storage process, which must recalculate the same value. To solve this, an in-memory cache system that allows specific values to be calculated only once but to be queried efficiently multiple times is used.

Market Execution Reading: similar to the previous problem, this responds to how another of the AT system modules is implemented. In particular, the problem is

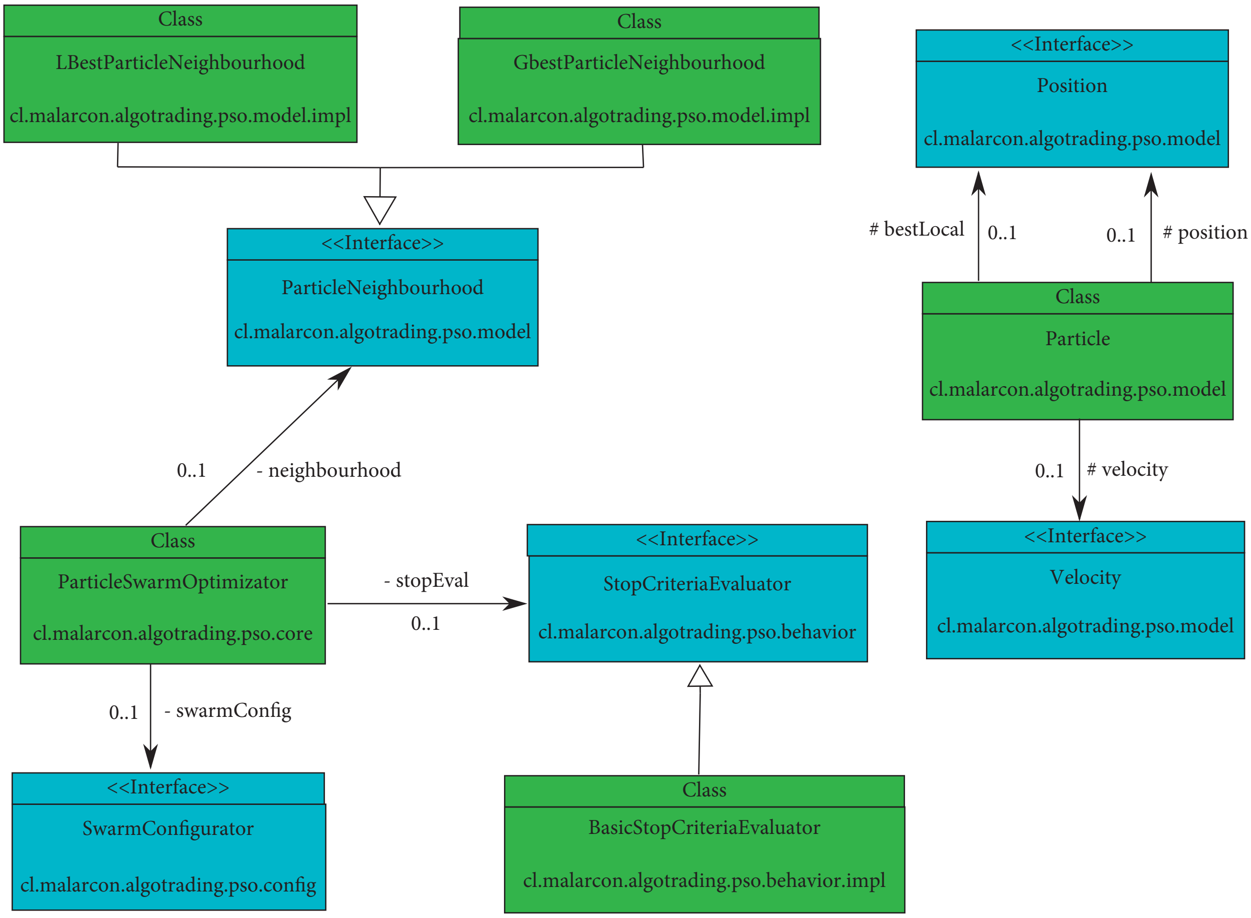

Class
LBestParticleNeighbourhood
cl.malarcon.algotrading.pso.model.impl
Class
GbestParticleNeighbourhood
cl.malarcon.algotrading.pso.model.impl
<<Interface>>
Position
cl.malarcon.algotrading.pso.model
# bestLocal
0..1
0..1
# position
<<Interface>>
ParticleNeighbourhood
cl.malarcon.algotrading.pso.model
Class
Particle
cl.malarcon.algotrading.pso.model
0..1
- neighbourhood
0..1
# velocity
Class
ParticleSwarmOptimizator
cl.malarcon.algotrading.pso.core
- stopEval
0..1
<<Interface>>
StopCriteriaEvaluator
cl.malarcon.algotrading.pso.behavior
<<Interface>>
Velocity
cl.malarcon.algotrading.pso.model
0..1
- swarmConfig
<<Interface>>
SwarmConfigurator
cl.malarcon.algotrading.pso.config
Class
BasicStopCriteriaEvaluator
cl.malarcon.algotrading.pso.behavior.impl

Figure 2: Full-implementation model of PSO. It does not include its own implementations of the problem to solve, since these are implemented in the automatic trader. It shows generic implementations of the neighborhood function (Local and Global Best) and the stopping criterion.

found in the market simulation routine present in OfflineCommunicationThread. Because this routine is based on the historical information of order executions, the relevant information must be loaded from a storage system (database). In the first implementation, each particle again loads the same data from the database for each iteration of PSO. This problem is solved using a shared cache of order executions that is used by all the particles in all their iterations.

4.6. System Testing

4.6.1. Initial Testing Version. The first experiment with the initial version is used to determine whether the system performs properly and is capable of generating positive returns. The specific values used in the experiment are as follows:

Period: 4 months (January–April 2012)

Instrument: LAN

Tick: 5 minutes

The period January–April 2012 is chosen because in that period, LAN has both increases and decreases in the share price. If only a period with increases is chosen, the risk management mechanism offered by the stop-loss band cannot be tested. However, if only a period with lows is chosen, the system will not perform the positioning (initial purchase), so it will remain inactive until a period of increase appears.

The results of experiment 1 are as follows:

Duration: 3.44 hours (12,388,202 ms)

Net returns: >0

MA Short: 30

MA Long: 58

StopLoss: 4,0543

StopWin: 11,3177

The experiment indicates that the process consumes a large amount of time due to the number of iterations performed and the size of the swarm. The positive aspect of these results is that there are gains at the end of the process, showing that the chosen parameters can be used to configure a trader that operates within a period reasonably close to the period of optimization.

Reviewing the values of the Stop-Loss and Stop-Win bands reveals a problem. The values exceed the maxima defined in the model, since the standard PSO velocity formula is applied. This indicates that an adjustment to the implementation of the formula must be made before proceeding with the final experiments.

The purpose of the second experiment is to review behavior and execution time for a shorter period. The input values are as follows:

Period: 2 months (January-February 2012)

Instrument: LAN

Tick: 5 minutes

The experiment aborts in the middle of the process because negative values are generated for the positions of the particles. This emphasizes the fact that the standard implementation of the velocity function cannot be applied to the AT model, so it must be adapted.

4.6.2. Optimized Version Testing. As mentioned, an optimized version of the AT system was generated. In this version, corrections to errors detected in the initial version of the system were implemented. The above experiment is repeated to determine the level of improvement introduced into the system. Thus, the specific values used in experiment 3 are as follows:

Period: 4 months (January–April 2012)

Instrument: LAN

Tick: 5 minutes

The results of experiment 3 are as follows:

Duration: 1 minute 26 seconds (85,950 ms)

Net returns: $294,186 (Profitability 69.96%)

MA Short: 45

MA Long: 52

StopLoss: 0

StopWin: 0.0040370020874758985

The experiment is successful, demonstrating that the AT system works properly within a reasonable time and that it is also able to find the parameters that allow positive profitability to be obtained for the analyzed period. Given this condition of increase, the PSO algorithm determined that under the conditions of the determined MA, it is more convenient to perform a large number of buy/sell operations, reflected by a Stop-Win band very close to zero. In addition, the algorithm determined that it is more advisable to use a zero risk to reduce losses. (See Table 2)

Finally, the experiment is executed 20 times to determine the best and worst times, together with the best net theoretical return. The best execution time obtained is 75,598 ms, and the worst is 102,842 ms; both values are well below the times obtained using the initial version of the system. The average execution time is 84,259.4 ms. The profitability of the best particles at the end of each PSO run fluctuates by an average of 438,870.6 CLP, with one particular iteration that uses a combination of parameters achieving a return of 138% on the first investment.

Table 2: Results of 20 executions of the AT model of experiment 3. The best time is only 75,598 ms, whereas the best theoretical net return is 690,757 CLP (profitability of 138%). The gains are due to the increasing trend for the period of the experiment; the parameters are adjusted accordingly by the PSO algorithm.

Time (ms)	Returns (CLP)	Short	Long	StopLoss	StopWin
78,892	382,304	42	48	0.004156852	0.00001
77,313	570,680	16	37	0.474434187	0.457211905
80,153	422,715	77	78	0.003876797	0.00001
102,842	327,270	101	128	0.008426064	0.00001
88,634	690,757	47	90	0	0.00001
94,216	351,071	70	71	0.007386603	0.00001
87,911	640,351	69	74	0	0.00001
78,439	647,545	35	39	0	0.00001
79,648	381,706	33	34	0.003689378	0.00001
79,306	404,035	1	33	0.00257833	0.00001
78,836	349,468	47	48	0.007894228	0.00001
84,482	384,969	60	106	0.003625605	0.00001
77,942	346,555	78	79	0.007217344	0.00001
94,406	373,798	1	170	0.004226087	0.00001
82,992	363,428	1	32	0	0.00001
89,537	355,787	126	127	0.003797629	0.00001
75,598	370,568	13	49	0.003803033	0.00001
76,913	651,456	20	28	0	0.00001
83,877	386,283	32	37	0.007186894	0.00001
93,201	376,666	3	150	0.00367221	0.00001

The experiment is repeated by varying the tick size. This allows the behavior of the AT system to be viewed in a manner that better approximates HFT. The specific values used in experiment 4 are as follows:

Period: 4 months (January–April 2012)

Instrument: LAN

Tick: 1 minute

In reviewing the results of the experiment shown in Table 3, the increase in execution time (at an average of 377,079 ms, equivalent to 6 minutes and 17 seconds) and the reduction of theoretical net return are notable. These changes are mainly caused by an increase in the Stop-Win band, which is the parameter that allows gains to be generated during a period of price increase. For many of the best results, we also calculate a Stop-Loss band greater than zero, indicating that the AT system will accept some level of risk to generate profits. This behavior may seem unfavorable in a period of sustained price growth, but it may be advantageous when there is price variation over very short periods.

5. Conclusions and Future Work

In the present research, we studied trading technologies that make it possible to operate under an HFT and/or an AT modality. We chose the statistical technique of MA for its simplicity, its ability to predict price trends based on the history of an instrument, and its applicability in optimization of techniques.

We reviewed information technologies that can be applied in conjunction with trading technologies, choosing

Table 3: Results of 20 runs of the AT model of experiment 4. The best time increases to 320,494 ms, while the best theoretical net return decreases to 285,351 CLP (profitability of 57%). The difference in behavior is due to the temporal granularity on which the system operates.

Time (ms)	Returns	Short	Long	StopLoss	StopWin
401,775	271,096	32	34	0	0.285258866
380,216	174,172	36	37	0	0.160024324
386,456	174,940	8	18	0	1.728610269
389,235	176,805	26	27	0	0.090294426
399,316	235,421	32	34	0	0.029627225
401,022	162,707	46	110	0	0.129552719
390,692	175,168	77	78	0	1.195936443
388,487	205,896	50	70	0.000384208	0.382646993
382,833	273,143	30	36	0	0.247501213
382,123	169,675	36	37	0	0.109817783
375,669	271,096	32	34	0	0.382314236
375,810	216,337	41	76	0.028521942	0.043130692
385,236	247,521	16	21	0	1.373017833
387,552	285,351	44	86	0.000293338	0.414043995
394,574	195,570	76	96	0.002536985	0.801816572
389,934	268,196	57	84	0	0.702944475
361,461	268,196	48	88	0	0.316077957
320,494	242,775	29	78	0.028291729	0.029311713
325,625	273,143	30	36	0	1.591565877
323,076	206,525	26	39	0.000372256	0.404751387

metaheuristics as the application for parameter optimization. Metaheuristics was chosen because a problem of profitability optimization in an equity market is an NP-class problem for which the application of search methods based on metaheuristics presents many advantages.

In addition, a design is presented for building an AT system based on the combination of trading and information technologies chosen. The design is sufficiently flexible to allow the system to be extended to other trading technologies and/or search solutions.

In the investigation, an initial version of the AT system is constructed under the proposed design. This allowed the first laboratory tests to be performed. These tests detected problems both with respect to the implementation of the AT system and with respect to special conditions that the PSO algorithm was not prepared to support. In particular, this version served to determine that in continuous but restricted domains the computation of PSO velocities must be bounded or modified in some way.

Finally, a second AT system is built based on the initial version but correcting the errors detected in the implementation of the AT model and applying the necessary limitations to the PSO algorithm. On the basis of the tests performed, it can be concluded that the defined AT system is capable of generating positive returns.

In this way, the chosen system corresponds to the improved version. Although the improved version is far from optimal, it provides a theoretical and practical basis for future research in a field in which the greatest amount of research comes from the private sector and not from the academic sector.

Regarding the application of PSO as an optimization algorithm, it is an effective solution for this problem type since it is able to optimize a set of disparate but bounded variables to a specific domain, thereby achieving a substantial improvement of the final solution. Regarding the application of PSO in optimizing the profitability of an AT system, it can be concluded that the velocity function must be altered or restricted depending on the trading model used.

How to obtain the optimal term of information prior to a given moment considered useful is a subject that remains to be studied in possible future work. One possible improvement would be to determine how changing the MA from simple to exponential would affect the optimal term. This would favor the recent trend of an instrument, ensuring that fluctuations that are too distant in time do not have undue importance in the model. Another future work would be the application of more complex AT system to the self-adjusting AT system, so that they include decision mechanisms with better risk management or that operate on smaller profit margins.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

Broderick Crawford was supported by Grant CONICYT/FONDECYT/REGULAR 1171243, and Ricardo Soto was supported by Grant CONICYT/FONDECYT/REGULAR 1160455.

References

[1] I. Aldridge, *High-Frequency Trading, A Practical Guide to Algorithmic Strategies and Trading Systems*, John Wiley & Sons, Inc, Wiley Trading Series, 2010.

[2] A. Pole, *Statistical Arbitrage, Algorithmic Trading Insights and Techniques*, John Wiley & Sons, Inc, Wiley trading series, 2007.

[3] C. Massey and G. Wu, "Detecting regime shifts: the causes of under- and overestimation," *Management Science*, vol. 51, no. 6, pp. 932–947, 2005.

[4] Economia and Negocios, "Bolsa responde a las críticas por fallas de su nuevo sistema de transacciones," 2010, http://www.economiaynegocios.cl/noticias/noticias.asp?id=73694.

[5] Bolsa de Comercio de Santiago, "Síntesis mensual Mayo 2010," 2010, http://www.bolsadesantiago.com/Sintesis%20y%20Estadisticas/S%C3%ADntesis%20Mensual%20Mayo%202010.pdf.

[6] W. Ketter, J. Collins, M. Gini, A. Gupta, and P. Schrater, "Detecting and forecasting economic regimes in multi-agent automated exchanges," *Decision Support Systems*, vol. 47, no. 4, 2009.

[7] F. Wang, K. Dong, and X. Deng, "Algorithmic trading systems: design and applications," *Frontiers of Computer Science in China*, vol. 3, no. 2, pp. 235–246, 2009.

[8] K. Kendall, *Electronic and Algorithmic Trading Technology*, Elsevier, Academic Press, Cambridge, MA, USA, 2007.

[9] R. Pardo, *The Evaluation and Optimization of Trading Strategies*, John Wiley & Sons, Inc, Wiley Trading Series, Hoboken, NJ, USA, 2008.

[10] J. Bigus, *Data Minig with Neural Networks*, McGraw-Hill, New York City, NY, USA, 1996.

[11] Advanced Trading and I. Schmerken, "Deutsche bank aims new stealth algo at buy-side," 2011, http://www.advancedtrading.com/algorithms/229401154.

[12] AMD WhitePaper, *Market Risk and Algorithmic Trading*, AMD & Toomre Capital Markets LLC, Phoenix, AZ, USA, 2007.

[13] M. Humphery-Jenner, "Optimal VWAP trading under noisy conditions," *Journal of Banking & Finance*, vol. 35, no. 9, pp. 2319–2329, 2011.

[14] S. Cook, "The complexity of theorem proving procedures," in *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, pp. 151–158, Shaker Heights, Ohio, USA, May 1971.

[15] R. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, Eds., pp. 85–103, Plenum, New York, NY, USA, 1972.

[16] L. Fortnow, J. Kilian, D. Pennock, and M. Wellman, "Betting Boolean-style: a framework for trading in securities based on logical formulas," *Decision Support Systems*, vol. 39, no. 1, pp. 87–104, 2005.

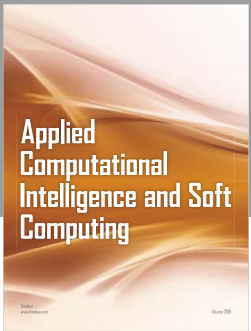
[17] K. Chang and A. Johnson, "Online and offline selling in limit order markets," in *Proceedings of the 4th International Workshop on Internet and Network Economics*, Shanghai, China, December 2008.

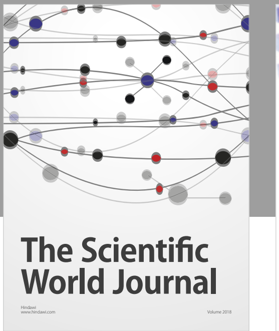
[18] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: overview and conceptual comparison," *ACM Computing Surveys*, vol. 35, no. 3, pp. 268–308, 2003.

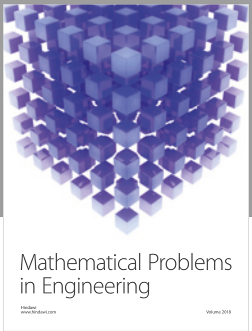
[19] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, Australia, November-December 1995.

[20] K. Jabri, D. Dumura, E. Godoya, A. Mouchetteb, and B. Bèleb, "Particle swarm optimization based tuning of a modified smith predictor for mould level control in continuous casting," *Journal of Process Control*, vol. 21, no. 2, pp. 263–270, 2011.

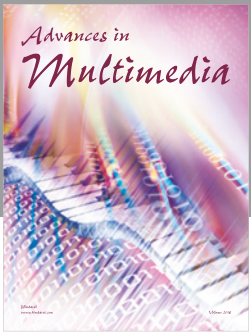
[21] M. Fikret, *Particle Swarm Optimization and Other Metaheuristic Methods in Hybrid Flow Shop Scheduling Problem*, Singapore Polytechnic School of Electrical and Electronic Engineering, Singapore, 2009.

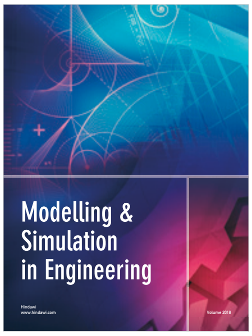

Applied Computational Intelligence and Soft Computing


The Scientific World Journal


Mathematical Problems in Engineering

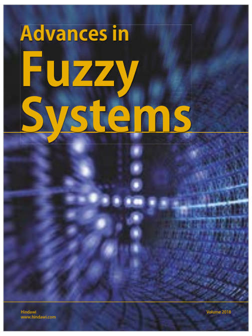
Journal of Engineering

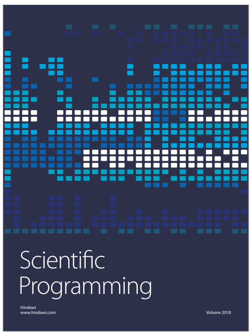

Advances in Multimedia

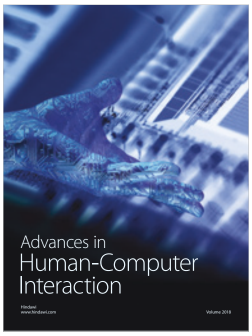

Modelling & Simulation in Engineering

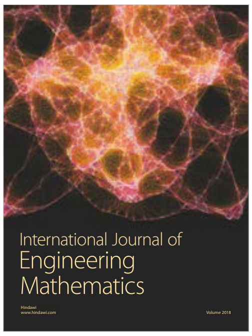
Advances in Artificial Intelligence

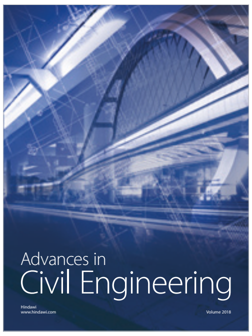
International Journal of Reconfigurable Computing

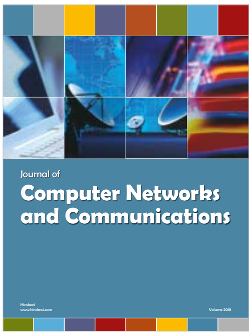

Advances in Fuzzy Systems


Scientific Programming


Advances in Human-Computer Interaction

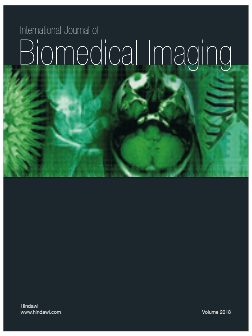

International Journal of Engineering Mathematics


Advances in Civil Engineering

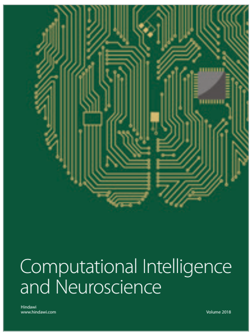

Journal of Computer Networks and Communications

Journal of Robotics

International Journal of Computer Games Technology


International Journal of Biomedical Imaging

Journal of Electrical and Computer Engineering


Computational Intelligence and Neuroscience