Knowledge-Based Systems 216 (2021) 106769

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Evolutionary ORB-based model with protective closing strategies

Mu-En Wu [a], Jia-Hao Syu [b], Jerry Chun-Wei Lin [c,*], Jan-Ming Ho [d]

[a] Department of Information and Finance Management, National Taipei University of Technology, Taiwan
[b] Department of Computer Science and Information Engineering, National Taiwan University, Taiwan
[c] Department of Computing, Mathematics, and Physics, Western Norway University of Applied Sciences, Norway
[d] Institute of Information Science, Academia Sinica, Taiwan

A R T I C L E I N F O

Article history:
Received 17 May 2020
Received in revised form 8 December 2020
Accepted 8 January 2021
Available online 2 February 2021

Keywords:
Opening range breakout
Genetic algorithm
Evolutionary computation
Protective closing strategies
Optimization

A B S T R A C T

Opening range breakout (ORB) is a well-known intraday trading strategy via technical analysis. ORB lacks robustness against market uncertainties (e.g., information from contradictory sources), and does not consider all relevant market characteristics. Furthermore, the closing strategies in generic ORB are not well defined. In this study, we developed an evolutionary ORB-based model, which utilized historical data to optimize thresholds in order to enhance profitability, and developed protective closing strategies aimed at to prevent unacceptable losses. Selecting appropriate thresholds and parameters for ORB is a non-trivial task, due to the fact that the search space exceeds sixty-five thousand options. We used evolutionary computation to derive rational strategies and parameters for ORB. The proposed framework based on a genetic algorithm optimizes the parameters related to threshold selection and protective closing strategies. In experiments, this resulted in annual returns of 9.3% (representing a 2.8% improvement over the original strategy) and Sharpe ratio of 2.5 (an improvement of 1.0), while reducing the maximum drawdown by half. The proposed scheme also reduced computational overhead by 89% compared to a grid search.

© 2021 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

Technical analysis and fundamental analysis are two major quantitative trading disciplines [1,2], which, unlike subjective sentiment and news-based strategies [3–5], use objective data to quantify trading patterns and generate trading signals. Fundamental analysis [6] estimates the market value of a company [7] by generating trading signals from financial statements indicating changes in income, cash flow, equity, and operating state. The low reporting frequency of financial statements means that fundamental analysis is best-suited to long-term investments with low trading frequency. Technical analysis [8] relies on analysis of statistical data gathered from trading activities [9] to identify and evaluate trading opportunities in an attempt to characterize patterns in price movements for use trading signals. Note that the amount of data available for technical analysis grows exponentially with trading frequency. Resolving difficulties in dealing with instances of high frequency trading [10] requires efficient algorithms based on computational intelligence [11] and big data [12]. In recent decades, technical analysis based on computational intelligence has become a popular trading discipline and topic of research.

Opening range breakout (ORB) is an intraday information-driven [13] momentum-type [14] trading strategy obtained via technical analysis. ORB is based on the assumption that the most dynamic period of any trading day is the first few minutes immediately after the market opens, which is referred to as the opening range. Information-driven trends that extend beyond the opening range can cause the price to break through a predetermined threshold. Holmberg et al. [15] reported that returns based on ORB strategies are higher than those obtained based on a normal distribution. In a report by Tsai et al. [16], ORB strategies achieved annual returns exceeding 8% in all five of the futures markets. Overall, ORB strategies have proven highly effective in theoretical analysis as well as practical applications based on empirical data.

Syu et al. [17] pointed out that ORB strategies do not fully account for historical data or market characteristics. This led to the development of threshold-adjusted ORB strategies (TA_ORB), in which prior information is used to define market characteristics and enhance profitability. They also reported that the win rate of TA_ORB is relatively low, due in part to the fact that ORB strategies always close the position when the market closes. In the event that contradictory information comes to light after taking a position, the conventional ORB closing strategy can lead to enormous losses. Essentially, ORB strategies lack robustness in the face of information from multiple sources or unforeseen market conditions. Our objective in this study was to develop

* Corresponding author.
E-mail addresses: mnwu@ntut.edu.tw (M.-E. Wu), f08922011@ntu.edu.tw (J.-H. Syu), jerrylin@ieee.org (J.C.-W. Lin), hoho@iis.sinica.edu.tw (J.-M. Ho).

https://doi.org/10.1016/j.knosys.2021.106769
0950-7051/© 2021 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

protective closing strategies aimed at enhancing the robustness of the ORB framework against uncertainties in the market.

Selecting the optimal parameters by which to formulate a strategy under ORB is a non-trivial issue. For example, using a grid search (i.e., exhaustive search) to derive the best parameters introduces the risk of overfitting [18]. Researchers have developed a number of parameter optimization tools, including simulated annealing [19], the gradient descent method [20], and genetic algorithm (GA) [21]. Simulated annealing is a probabilistic technique (inspired by the hardening process used in metal work) used to identify the optimal combination of objective functions [19]. Gradient descent is an optimization algorithm that iteratively moves in the direction of the steepest descent as determined by a gradient [20]. However, an uneven solution space with multiple peaks cannot guarantee a gradient indicating the correct direction. GA is a heuristic approach in evolutionary computation [22], which has proven highly effective in nonconvex multi-peak optimization problems [23]. This approach is based on the concept of survival by natural selection [24], in which a strong individual (a strong parameter set) has a higher likelihood of survival. Furthermore, the process of selection, crossover, and mutation leads to a wide range of potential solutions. These advantages motivate us employed a GA to facilitate the selection of parameters for our ORB-based model.

In this study, we sought to establish ORB strategies capable of improving trading performance and computational efficiency. The proposed protective closing strategies alert investors of situations requiring their attention, while stop-loss and take-profit mechanisms help to prevent unbearable losses and realize profit [25]. We developed a GA-based ORB framework (GAORB) to overcome the complexity inherent in the selection of parameters. Two variants of the scheme based on different selection criteria (respectively referred to as GAORB_Ret and GAORB_Sharpe) allow investors to focus on different indicators, such as risk or profit. GAORB is applicable to many sub-algorithms used for the selection of indicators and the development of protective closing strategies, including GA_Ret, GA_Sharpe, GA_Ret_SL, and GA_Sharpe_SL.

In experiments, GA_Ret and GA_Sharpe improved on the original ORB by increasing annual returns by 1.667% (to 8.068% and 8.305%) and the Sharpe ratio by 0.365 (to 1.810 and 1.865), while reducing computational overhead by 60.9%. GA_Ret_SL and GA_Sharpe_SL improved annual returns by 2.667% (to 9.071% and 9.303%) and the Sharpe ratio by 0.926 (to 2.302 and 2.495), while reducing computational overhead by 89.1%. Overall, our results demonstrate that the proposed threshold adjustment can enhance profitability, and the proposed stop-loss mechanism can help to stabilize performance. The GAORB framework was also shown to improve the efficiency and rationality of parameter selection. This paper makes the following contributions as:

1. We first developed an ORB-based model that fully takes historical data or market characteristics into account for improving trading performance.
2. The developed a ORB-based model facilitates the selection of parameters using the genetic algorithm to achieve good performance.
3. Several protective closing strategies were proposed to stabilize and enhance profits in the developed model.

2. Literature review

2.1. Original opening range breakout

Opening range breakout (ORB) is an event-driven, intraday trading strategy based on the concept of momentum. ORB detects pricing trends for the generation of trading signals, while setting upper and lower price thresholds based on the highest and lowest prices within the opening range, which is the period shortly (15 to 30 min) after the market opens. As long as no unforeseen events occur during that day, prices are expected to fluctuate between the two thresholds. In the event of newsworthy changes, price can be expected to break out of the thresholds and generate trading signals. Based on the concept of momentum, if the price breaks out beyond the upper threshold, then the ORB will generate a long (buy) signal. If the price breaks out beyond the lower threshold, then the ORB will generate a short (sell) signal. As an intraday strategy, ORB closes (clears) all positions at the close of the market.

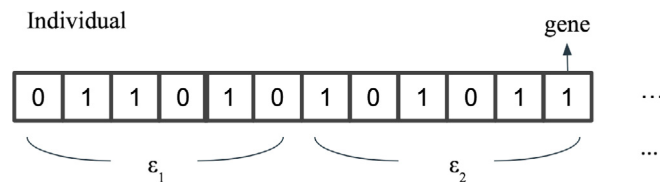

Fig. 1. GA encodes the parameters into binary gene.

Holmberg et al. [15] presented an ORB strategy based on normally distributed returns, which produced returns significantly higher than zero in conventional statistical tests. In seeking to determine the best length for the open range in each market, Tsai et al. [16] developed an ORB strategy that achieved annual returns of over 8% on the DJIA, NASDAQ, S&P, HSI, and TAIEX. Syu et al. [17] reported that the ORB strategy was profitable when applied to the Taiwanese financial market; however, the ORB does not take historical data or market characteristics fully into account. ORB strategies have also been developed with adjusted trading thresholds to account for market characteristics with the aim of improving profits and reducing risk. One such model achieved annual returns of 32.25% between 2008 and 2012, which is 2.6 times the original ORB. The ORB studies mentioned above succeeded in generating trading signals indicating a suitable time to enter the stock market, but the closing mechanism involves clearing the position when the market closes. In cases of unreliable signals or contrary events, the ORB can lead to enormous losses. ORB strategies clearly require a reliable closing mechanism.

2.2. Genetic algorithm

Genetic algorithm (GA) was first introduced by Holland in 1975 [26], which is a subset of evolutionary computation [22] that uses metaheuristic techniques for solving optimization problems [27]. GA is inspired by the theory of evolution [28] by natural selection [24], based on the principle that the genes that give strong individuals a higher likelihood of survival can be passed on by parents to the next generation. The process of evolution involves three operations: selection, crossover, and mutation [21, 29].

A chromosome (or an individual) in GA represents a set of solutions, which can be encoded in binary format (0 and 1), as shown in Fig. 1. The number of individuals in a generation is referred to the population size, and each individual is evaluated in terms of a fitness function. A chromosome with a higher fitness value has a higher probability of surviving into the next generation. A selection operator selects the individuals for the next generation based on their calculated fitness values, using methods such as Roulette Wheel Selection [26], Stochastic Universal Sampling [30], or Save-best [31]. Roulette wheel selection [32] determines survivors in terms of probability proportional to the fitness value, while giving even weak individuals a small chance of surviving. Save-best selection allows the survival only of individuals with the highest fitness values. Stochastic universal sampling [30] ensures a selection of novel offspring to enhance the variety of the population. Crossover and mutation are the evolutionary processes driving the creation of the next generation.

GA has proven highly effective in resolving NP-hard optimization problems. It has been applied in many domains, such as the traveling salesman problem [33,34], rule mining [35], energy [36, 37], and finance [38–41]. Metawa et al. [38] solved the issue of banking decisions in credit crunch environment by adopting GA to maximize profits and minimize the risk of default. Núñez-Letamendia [39] applied a GA-based model to deal with technical trading systems in which the complexity exponentially grows with the number of technical rules and decisions. Díaz et al. [40] sought to resolve the problem of multi-period index tracking by developing a hybrid model in which the GA was used to select stocks for the creation of an index tracking portfolio. Srinivasan and Kamalakannan [41] used multi-objective GA-based model to assist the financial risk management through the analysis of data pertaining to credit cards and credit applications.

2.3. Performance indicators

Trading strategies can be evaluated in terms of profitability and/or risk [42]. In measuring profitability, the total profit, annual returns, and the win rate are generic measures used for evaluation. The commonest indicator for measuring profitability is total profit, which refers to the revenue attributable to a given strategy in a given trading period. Annual returns refers to the returns generated by a given strategy over the period of a year [43], which can be calculated by dividing total profit by the average costs during the trading period, the result of which is divided by the number of the years. Win rate refers to the number of wins divided by the total number of transactions, indicating the probability that a given strategy will make a profit.

The Sharpe ratio [44–47], profit factor (PF) [48–50], and maximum drawdown (MDD) [51,52] are general terms used in the evaluation of risk. The Sharpe ratio is derived by dividing excess annual returns by the standard deviation of returns, as shown in Eq. (1), where R is the annual returns attributable to the strategy, r_f is the risk free rate, and σ is the annualized standard deviation of daily returns. Note that the effect of r_f is usually disregarded. The Sharpe ratio is used to indicate the trade-off between profitability and volatility; i.e., how much profit can be earned under a unit of risk (volatility) [53]. Generally, a higher Sharpe ratio is more attractive to investors. PF indicates the net profits divided by the absolute value of the net loss [48], indicating the amount of profit that can be earned for each dollar lost. A PF value exceeding 1 indicates that a given strategy is profitable, whereas a PF value below 1 indicates that total losses exceed total profits MDD indicates the maximum observed loss associated with a given portfolio between a peak and a trough (i.e., before a new peak is attained) [54], indicating the maximum loss that could be incurred during trading. A smaller MDD indicates better risk management performance.

$$\text{Sharpe Ratio} = \frac{R - r_f}{\sigma} \approx \frac{R}{\sigma} \tag{1}$$

3. Proposed GAORB framework and protective closing strategies

In this section, we outline the three main components of the proposed framework, which is the threshold adjustment ORB strategy and protected closing strategies optimized by GA. Section 3.1 introduces the threshold adjustment ORB strategy, which also illustrates the trading mechanism of the proposed framework. Section 3.2 presents the protected closing strategies designed for ORB strategy to reduce the trading risk. With the description of trading and protected closing strategies, the parameters of the proposed framework are listed and optimized in Section 3.3. Details of each components of the developed GAORB are then illustrated below.

3.1. Threshold adjusting ORB strategy (TA_ORB)

Syu et al. [17] proposed a modified ORB strategy, referred to as threshold adjusting ORB (TA_ORB) to enable the efficient utilization of historical information and market characteristics, which is applied in the our framework. This involves adjusting the upper and lower price thresholds (h and l) via ε (ε_1 and ε_2) and σ to create new upper and lower bounds (B_u and B_l), which can be formalized using Eq. (2) (see Fig. 2). σ indicates the standard deviation of the price within the opening range, which is a varied from day to day. ε_1 and ε_2 indicate the scale of price threshold adjustments, which is a discrete parameter. TA_ORB generates trading signals based on the new upper and lower bounds, which means it obtains the information of h, l, ε, and σ within the opening range to calculate the upper and lower bounds (B_u and B_l). After the opening range, if the price breaks out beyond the upper (below the lower) bound, TA_ORB then generates a long (short) signal, and buy (sell) a position of the futures.

As shown in Fig. 3, TA_ORB uses a moving window [55] for the parameter selection of ε with the highest returns (or highest Sharpe ratio) in the previous two months of training data to be applied to the testing data for the following months. For example, as for trading in March 2018, the framework will use the data from January 2018 to February 2018 as training data, and find the ε_1 and ε_2 to optimize the trading return (Sharpe ratio) within the training data. The selected ε_1 and ε_2 are implemented in March 2018 to define the bounds and generate trading signals. The training and testing windows move forward by one month at a time. To facilitate calculation, we avoided calculating the leverage of futures contracts by purchasing and selling futures at the market price. Thus, we derived the profits from the difference between the purchase and selling prices with the unit of point (a point is 200 new Taiwan dollars by the futures contracts), instead of calculating the margin and leverage. Furthermore, we purchased the same number (called a unit or a position) of futures every day regardless previous profits and losses.

$$\begin{aligned} B_u &= h + \varepsilon_1 \times \sigma \\ B_l &= l + \varepsilon_2 \times \sigma \end{aligned} \tag{2}$$

In this paper, the parameters used in TA_ORB (ε_1 and ε_2) are optimized in accordance with the proposed GAORB framework. Syu et al. [17] discovered ε only at coarse resolutions; however, each grid used in the search is likely to exceed 1, which means that many potential solutions must be abandoned. Thus, precision can be improved by obtaining ε from data of higher resolution.

As shown in Fig. 4, we investigated the size distribution of σ to determine the optimal search resolution, eventually settling on the 99th percentile. The highest 1% of σ was 21.8 points (the unit used in this paper); therefore, we set the jump size (search resolution) to $\frac{\sigma}{32}$ ($2^{-5}\sigma$) in order to ensure that each jump size was less than 1 and would cover almost all possible solutions. The search range of ε was from -σ to σ in increments of $\frac{\sigma}{32}$. This resulted in 64 candidate values for ε, which are encoded in 6 bits for both ε_1 and ε_2. The GA-based ORB with threshold adjustment is termed GAORB. We also developed two variants based on different selection criteria (respectively referred to as GAORB_Ret and GAORB_Sharpe), to allow investors to focus on different indicators, such as risk and profit.

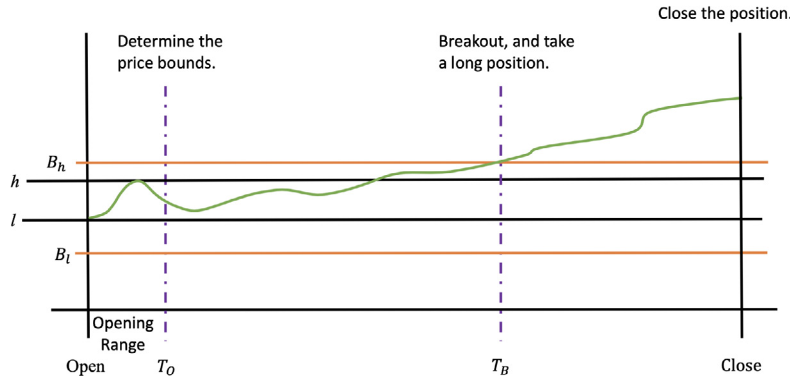

Fig. 2. TA_ORB strategies; x- and y-axis are the futures price and the time axis.

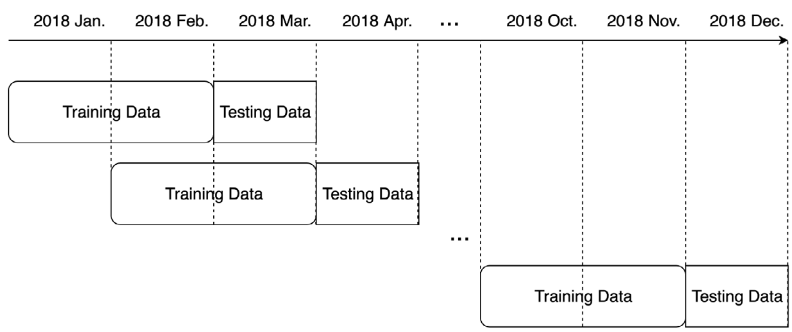

Fig. 3. The moving window mechanism.

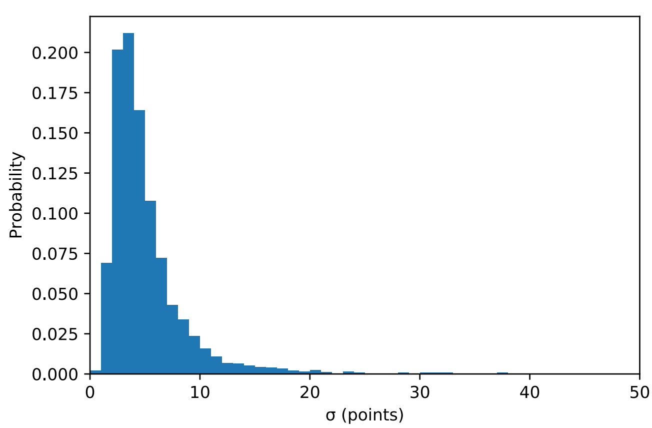

Fig. 4. Distribution of the σ size from 2007 to 2018; x- and y-axis represent the probability and the size of σ (standard deviation of price in opening range).

3.2. Protective closing strategies

The protective closing strategies (stop-loss and take-profit) are independent mechanisms intended to prevent unbearable losses, and are applicable to GAORB_Ret as well as GAORB_Sharpe. Under the stop-loss mechanism, all positions are immediately closed if the unrealized loss exceeds threshold (SL). Consider the example where SL is set to 100 points, if we take a long position at 1000 points and the price dropped to below 900 points, then the stop-loss mechanism would be triggered, such that the position would be closed immediately (i.e., the unrealized loss is greater than SL).

Under the GAORB strategy, situations where the unrealized loss is greater than the difference between the upper and the lower bounds $(B_u - B_l)$ showed that the opposite position should be taken (i.e., ORB made an erroneous decision) and the current position should be stopped-loss. Therefore, we set the stop-loss threshold (SL) to be $T_{SL} \times (B_u - B_l)$ in the proposed GAORB. Note that T_{SL} is assigned by one of four discrete values, such as $\frac{1}{3}$, $\frac{2}{3}$, 1, and ∞, and ∞ ensures that the stop-loss mechanism is not triggered.

The take-profit mechanism is used to ensure that a profit is realized. If the unrealized gain is greater than or equal to threshold TP and the price drawdown from the highest price exceeds the ratio of maximum unrealized gain (R_{DD}), then all of the positions are immediately closed. To be brief, once the gain reaches the threshold TP, the take-profit mechanism is activated. After that, once the price drops below (highest price $-R_{DD}\times$ maximum unrealized gain), all of the positions are immediately closed. Consider an example where R_{DD} is set to 20% and TP is set to 100 points. If we take a long position at 1000 points and the price increases to 1200 points, the take-profit mechanism is activated since the unrealized gain is larger than TP, 100 points. After 1200 points, the price began to fall. If the price drops below to $1200 - 20\% \times 200 = 1160$, such that the position would immediately be closed (i.e., drawdown from the highest price exceeds $R_{DD}\times$ maximum unrealized gain $= 20\% \times 200 = 40$ points).

Also, we set the take-profit threshold (TP) to be $T_{TP} \times (B_u - B_l)$ in the proposed GAORB. Therefore, the take-profit mechanism includes two parameters: T_{TP} and R_{DD}. This process is simplified by setting T_{TP} at the value of T_{SL}, indicating that the unrealized gain or loss is less than T_{SL}, such that protective closing strategies are not to be activated. R_{DD} was set at discrete values of $\frac{1}{3}$, $\frac{2}{3}$, 1, or ∞, in which ∞ indicates that the take-profit mechanism will never be triggered.

We also employed a moving window mechanism [17] for the selection of parameters from training data covering the last two months. The parameters are then applied to testing data for the following month with the window sliding one month for each

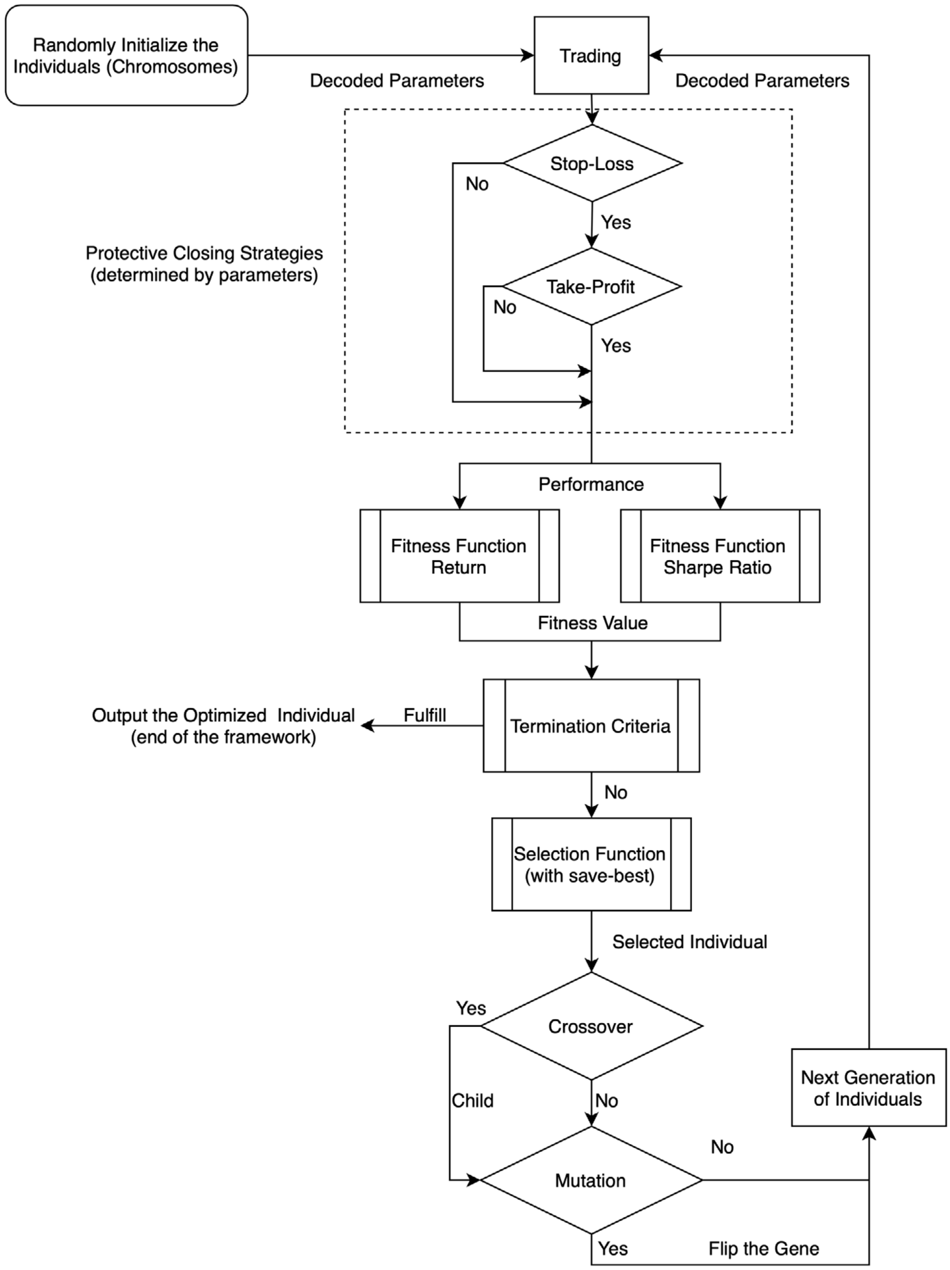

Fig. 5. The flowchart of the designed GAORB framework.

period, as shown in Fig. 3. The parameters in this paper included ε_1, ε_2, T_{SL}, and R_{DD}, which resulted in 2^{16} parameter combinations. Under these circumstances, using the grid search approach to find optimal solutions would be a non-trivial task. We therefore developed a GA-based framework called GAORB to optimize the parameters used in ORB.

3.3. Proposed GAORB framework

The proposed GAORB framework is illustrated in Fig. 5. GAORB uses data from the previous two months to derive appropriate parameters for each testing month.

First, we randomly generate individuals for the initial generation, and each generation (population size) contains 25 individuals . As shown in Table 1, the chromosomes of individuals are encoded in binary format. The 1st to 6th bits represent ε_1 and the 7th to 12th bits represent ε_2; both of them include 6 digits with the range from 0 to 63. We then encode ε_1 and ε_2 from [$\frac{-32}{32}$, $\frac{-31}{32}$, …, $\frac{31}{32}$] to [0, 1, …, 63], which covers approximately -1 to 1, increased by $\frac{1}{32}$. The 13th to 14th bits represent the T_{SL} and the 15th to 16th bits represent R_{DD}; both of them encode [$\frac{1}{3}$, $\frac{2}{3}$, 1, ∞] into the binary form [0, 1, 2, 3], which are used to determine whether the implementation of protective closing strategies is required. Thus, if the T_{SL} and R_{DD} are ∞, then no protective closing strategy is implemented.

After the initial encoding progress, GAORB is started to run the evolutionary process. For each iteration, individual chromosomes are decoded into a parameter set, and the proposed fitness functions are then used for evaluation (i.e., **trading return** or **Sharpe ratio**). Strategies that use the **trading return** as a fitness function are referred to as GAORB_Ret, whereas strategies that use the **Sharpe ratio** as a fitness function are referred to as GAORB_Sharpe.

Algorithm 1: Developed GAORB model

for *Each testing month* **do**
  Implement GAORB model to find the optimized ORB parameters in the past two-month training data:
  Randomly generate individuals for initial generation
  for *Repeat* $N_{generation}$ *times* **do**
    Calculate the *fitness value* for each individual from the trading return or Sharpe ratio
    Select individuals by *fitness value*
    for *Each individual* **do**
      if *Crossover* (P_{cross}) **then**
        Randomly select another parent
        Child gene picked from parents
        Child replaces the parent
    for *Each gene of each individual* **do**
      if *Mutate* (P_{mutate}) **then**
        Flip that gene (0 to 1, vice-versa)
    if *Termination criterion is satisfied* **then**
      Return the best individual, and terminate the progress
  Trade the ORB strategies in testing data by the best parameters from the derived final solutions (the individual)

Eqs. (3) and (4) respectively present the two sigmoid fitness functions for GAORB_Ret and GAORB_Sharpe (see Fig. 6), where x is the function input equal to the trading returns or Sharpe ratio. The sigmoid function [56] is monotonic and bell-shaped in the first derivation, as it matches the distribution of returns, which makes it suitable for the proposed model. The output of the fitness function is the fitness value representing the survival probability of the individual. Thus, the probabilities of obtaining returns of 0% and 5% are respectively 50% and 75%, where 5% is the expected returns based on the market trend. The probabilities of obtaining a Sharpe ratio of 0 or 0.6 are respectively 50% and 75%, where 0.6 is the expected Sharpe ratio based on the market trend. Using the fitness functions defined above, the survival probability of an individual with 5% returns (Sharpe ratio of 0.6) has half more probability than that of an individual with 0% returns (Sharpe ratio of 0).

$$FF\_Ret(x) = \frac{1}{1 + e^{-20 \ln 3 \times x}} \tag{3}$$

$$FF\_Sharpe(x) = \frac{1}{1 + e^{-1.67 \ln 3 \times x}} \tag{4}$$

After the evaluation progress of each individual, the selection operator selects survivors with great fitness value for next generations. Three commonly-used methods such as roulette wheel selection [26], save-best selection, and stochastic universal sampling [30] are used for the selection operation. Note that each method is used to contribute one-third individuals of a population. In addition, the crossover rate (P_{cross}) is initially set at 80% for evolution. Each survivor has P_{cross} probability to be randomly selected by performing the crossover operation for next generation. Furthermore, the mutation rate (P_{mutate}) is initially set at 5% for evolution. Each gene of a survivor has P_{mutate} probability to run the mutation operation, which is to switch the value from 0 to 1, an vice versa. Two termination criteria for the developed GAORB are: (1) if the number of iterations exceeds 100, the algorithm is then terminated; or (2) if the individuals are converged for 50 generations, the algorithm is then terminated. After that, the individual with the highest fitness value is considered as the solution. This output is then decoded into a parameter set for the ORB strategies and will be examined in the test dataset. The pseudo-code of the designed GAORB is presented in Algorithm 1.

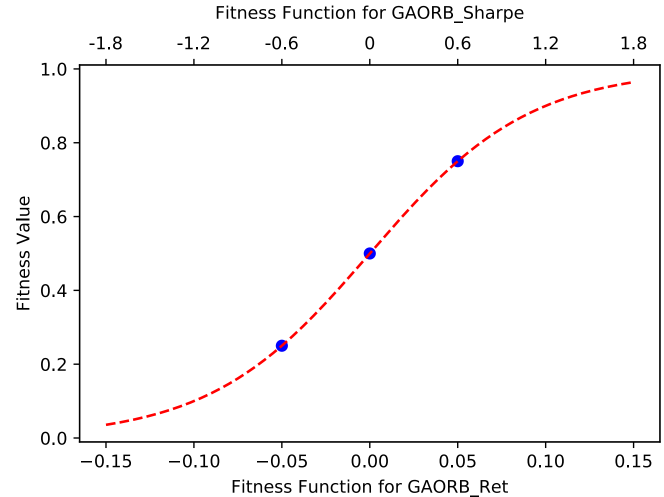

Fig. 6. Fitness functions for each GAORB. The x-axis represents the trading return and the Sharpe ratio, and the y-axis represents the fitness value of the corresponding return for Sharpe ratio.

Table 1
Encoding the chromosomes.

1–6 bits, 7–12 bits	ε_1, ε_2	13–14 bits, 15–16 bits	T_{SL}, R_{DD}
000000	$-\frac{32}{32}$	00	$\frac{1}{3}$
000001	$-\frac{31}{32}$	01	$\frac{2}{3}$
000010	$-\frac{30}{32}$	10	1
000011	$-\frac{29}{32}$	11	∞
⋮	⋮		
111110	$\frac{30}{32}$		
111111	$\frac{31}{32}$		

Table 2
The naming rules of GA-based model.

GAORB	Return	Sharpe ratio
Without closing strategy	GA_Ret	GA_Sharpe
With stop-loss	GA_Ret_SL	GA_Sharpe_SL
With stop-loss and take-profit	GA_Ret_SLTP	GA_Sharpe_SLTP
Collectively called	GAORB_Ret	GAORB_Sharpe

4. Experiment results

This section outlines the trading performance, time efficiency, convergence, and effectiveness of the GAORB strategies. The naive GA-based models without the proposed closing strategies are referred to as GA_Ret and GA_Sharpe. GA-based models with the stop-loss mechanism are referred to as GA_Ret_SL and GA_Sharpe_SL. GA-based models with the stop-loss as well as the take-profit mechanisms are referred to as GA_Ret_SLTP and GA_Sharpe_SLTP. The naming rules of the GA-based models are listed in Table 2, where columns indicate the fitness function, and rows indicates the protective closing strategy. In addition, we adopt the original ORB strategy as a benchmark to compare with the proposed evolutionary ORB-based models. The benchmark strategy in all of the figures and tables is the original ORB strategy, which does not adjust the price threshold and without protective closing strategies. All of the GA-based models are referred to as GAORB, and the models with fitness functions based on returns and Sharpe ratio are respectively referred to as GAORB_Ret and GAORB_Sharpe.

From Figs. 7 to 9, the x-axis indicates the trading day and the y-axis indicates the cumulative profits. The vertical black lines

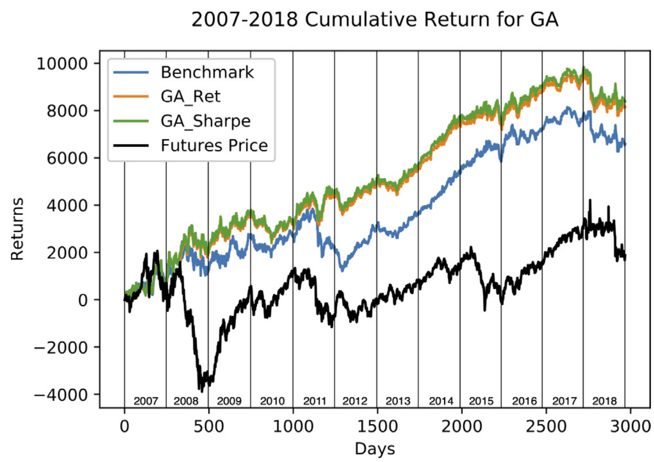

Fig. 7. Cumulative profit of benchmark, GA_Ret, and GA_Sharpe.

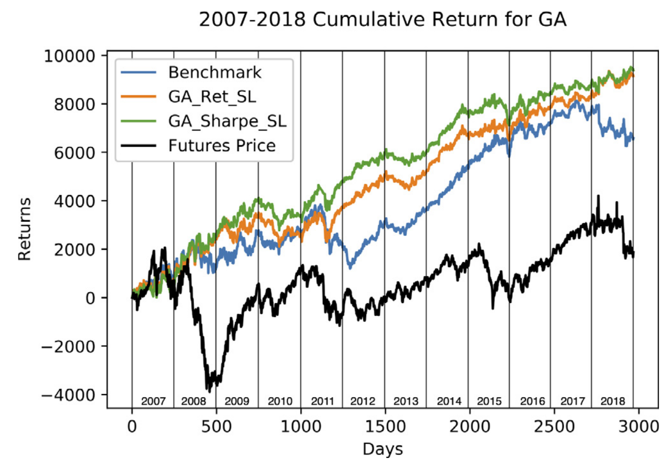

Fig. 8. Cumulative profit of benchmark, GA_Ret_SL, and GA_Sharpe_SL.

Table 3
Performance of strategies.

	Benchmark	GA_Ret	GA_Sharpe
Total profit	6579	8141	8380
Annual returns	6.520%	8.068%	8.305%
Win rate	50.475%	51.067%	51.304%
Sharpe ratio	1.473	1.810	1.865
Profit factor	1.095	1.118	1.122
MDD	2638	1824	1780

Table 4
Performance of strategies with stop-loss mechanism.

	Benchmark	GA_Ret_SL	GA_Sharpe_SL
Total profit	6579	9154	9388
Annual returns	6.520%	9.071%	9.303%
Win rate	50.475%	42.756%	39.017%
Sharpe ratio	1.473	2.302	2.495
Profit factor	1.095	1.158	1.177
MDD	2638	1401	1336

indicate the beginning and end of the year from 2007 to 2018. The blue curve indicates the cumulative profits obtained using the original ORB strategy (the benchmark in this paper), and the black curve indicates the price of Taiwan Index Futures (TXF), which is the commodity in which we invested. Note that the blue and black curves are the same in Figs. 7 to 9 because of the same indicator lines and benchmark. The orange curve indicates the results obtained using GAORB_Ret and the green curve indicates the results obtained using GAORB_Sharpe.

In the following experiments, TXF data provided by the Taiwan Futures Exchange were used as a trading dataset. The 1-minute OHLC data (opening, highest, lowest, and closing prices) covered the period from 2007/11/1 to 2018/12/31, including bull and bear markets. The dataset included approximately 10^6 of OHLC data.

4.1. Trading performance of GAORB strategies without protective closing strategies

Table 3 lists the performance indicators used to assess GA_Ret and GA_Sharpe, both of which lack a protective closing strategy. GA_Ret and GA_Sharpe significantly outperformed the benchmark in terms of **total profits** (+1681.5 averagely), **annual returns** (+1.667% averagely) and **Sharpe ratio** (+0.365 averagely), indicating they generated more profit in a more stable manner. GA_Ret and GA_Sharpe provided the best results for **profit factor** and **win rate**. Note that the **MDD**s of GA_Ret and GA_Sharpe were 30% lower than that of the benchmark, indicating that the potential risk was far lower.

The cumulative profit curves in Fig. 7 show that GA_Ret and GA_Sharpe (orange and green curves) continuously grew to reach new heights. They also avoided the sharp decline of roughly 2000 points in 2011, which would otherwise have consumed all of the profits from the previous three years. Note that all three of the strategies incurred huge losses associated with the trade war between the United States and China in 2018. This is a clear indication of the need for protective closing strategies to prevent unpredictable losses.

4.2. Trading performance of GAORB strategies with protective closing strategies

Fig. 8 and Table 4 present the cumulative profits and performance indicators for the benchmark, GA_Ret_SL, and GA_Sharpe _SL. As shown in Table 4, GA_Ret_SL and GA_Sharpe_SL outperformed the benchmark in terms of **annual returns** (+2.667% on average) and **Sharpe ratio** (+0.926 on average) with lower risk in terms of **MDD** (−50% on average), which meant they reduced the risk by half and increased the profits. As shown in Fig. 8, GA_Ret_SL and GA_Sharpe_SL (orange and green curves) grew more steadily than the curves in Fig. 7. GA_Ret_SL and GA_Sharpe_SL also prevented enormous losses in 2018, which provided positive returns even during the trade war. Nonetheless, the **win rate** dropped to 43% and 39%, due to the fact that the stop-loss mechanism increased the likelihood that loss positions would be closed (i.e., the possibility of a recovery was abandoned). In general, stop-loss mechanisms increase the likelihood of loss events, but significantly reduce their magnitude. Thus, despite the fact that the stop-loss mechanism lowered the win rate, it also produced slight increase in profitability by providing a considerable improvement in stability. To overcome the negative impact of the stop-loss mechanism, we implemented the take-profit mechanism to realize profits earlier and further increase the win rate.

Fig. 9 and Table 5 present the cumulative profits and performance indicators from the benchmark, GA_Ret_SLTP and GA_ Sharpe_SLTP. Overall, the implementation of a take-profit mechanism led to sharp drop in performance to below that of models with only a stop-loss mechanism. Only GA_Ret_SLTP outperformed the benchmark, and then only slightly in terms of **annual returns**, **Sharpe ratio**, and **profit factor**, while reducing **MDD**

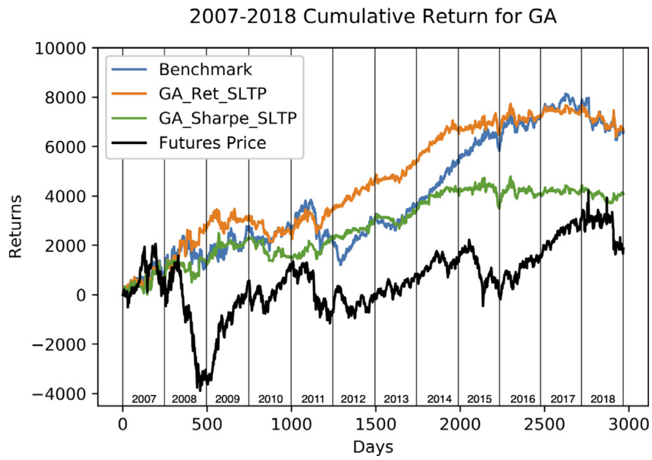

Fig. 9. Cumulative profit of benchmark, GA_Ret_SLTP, and GA_Sharpe_SLTP.

Table 5
Performance indicators of the strategies with stop-loss and take-profit mechanisms.

	Benchmark	GA_Ret_SLTP	GA_Sharpe_SLTP
Total profit	6579	6671	4089
Annual returns	6.520%	6.610%	4.051%
Sharpe ratio	1.473	1.756	1.320
Win rate	50.475%	44.444%	43.578%
Profit factor	1.095	1.118	1.09
MDD	2638	1337	1249

by half. The orange curve in Fig. 9 (GA_Ret_SLTP) is smoother than the blue curve but ends at nearly the same point. The performance of GA_Sharpe_SLTP was slightly worse than that of the benchmark, resulting in far lower performance indicators and **MDD** cut by half. The win rates of GA_Ret_SLTP and GA_Sharpe_SLTP slightly were better than those of GA_Ret_SL and GA_Sharpe_SL, but still below the benchmark. As shown in Fig. 9, GA_Sharpe_SLTP presented less fluctuation (greater stability); however, it eliminated the possibility of attaining large growth. Overall, the take-profit mechanisms slightly increased the win rate and stability, but severely limited the profits and eliminated the possibility of a sharp rise. We do not recommend the implementation of take-profit mechanisms with the proposed model.

4.3. Convergence of GAORB

GA proceed through an iterative evolutionary process, which means that the termination criteria must be set carefully to maximize performance and save computing power. If the criteria were too strict, then the GA would run too long and increase the cost of computing. If the criteria were not strict enough, then the GA would stop too soon and in so doing fail to find an appropriate solution. Thus, we conducted an experiment aimed at assessing the termination criteria based on the convergence of the GA.

The termination criteria in this paper included a failure to improve the current individual in 50 consecutive generations, and a maximum of 100 iterations. Based on observations of convergence steps (C.S.) in each trading month and each GAORB_Ret strategy, we calculated the average and standard deviation (Std) of the C.S., as shown in Table 6 (10 repetitions). The C.S. values of GAORB_Ret and GAORB_Sharpe were very close, showing a gradual increase with the size of the solution space with the application of more mechanisms.

Table 6
Convergence step for GAORB strategies.

GAORB	Average C.S.	Std of C.S
GA_Ret	63.860	12.245
GA_Ret_SL	70.858	15.015
GA_Ret_SLTP	73.813	15.799
GA_Sharpe	64.271	12.548
GA_Sharpe_SL	71.215	14.701
GA_Sharpe_SLTP	75.307	15.690

Table 7
Time efficiency of GAORB.

GAORB	Avg. C.S.	Avg. searched parameter sets	Computation reduction
GA_Ret	63.860	1597	61.0%
GA_Ret_SL	70.858	1771	89.2%
GA_Ret_SLTP	73.813	1845	97.2%
GA_Sharpe	64.271	1607	60.8%
GA_Sharpe_SL	71.215	1780	89.1%
GA_Sharpe_SLTP	75.307	1883	97.1%

Fig. 10 illustrates the convergence of GAORB_Ret for Jan. 2007, which was the first month covered by the data. The y-axis indicates the trading returns, and the x-axis indicates the specific iteration. The black spots indicate the top-5 individuals in each generation, and the red line connects the best individuals of each generation. As mentioned in Section 3.3, the selection function saved the best 33% of the individuals in each generation, which allowed for a constant increase in performance that can be observed in Fig. 10. From Fig. 10, it is obvious to see that the average convergence step is about 60 to 70 steps. Subtracting the 50-step unchanged termination criterion, it indicates that the proposed GA-based model can be converged within 10 to 20 steps with sufficient exploitation and at robustness result in average. Furthermore, we found that most of the improvements occurred within the first 10 generations. We also found that if the best individual remained unchanged for 20 consecutive generations, then the GA had likely reached convergence. This is a clear indication that the termination criteria (non-improvement for 50 consecutive generations) was sufficiently strict.

4.4. Time efficiency of GAORB

In the experiments, the chromosomes used 12, 2, and 2 bits to represent the parameters of threshold adjustment, stop-loss mechanism, and take-profit mechanism, respectively. The search space included a total of approximately 2^{16} parameter combinations. Thus, we employed GAORB to speed up computation, the time efficiency of which is shown in Table 7 (includes average convergence steps, average searched parameter sets, and the computation reduction).

The results in Section 4.1 show that GA_Ret converged at an average of 63.860 steps, which means that it calculated only $63.860 \times 25 = 1597$ sets of parameters among the 2^{12} sets (i.e., a reduction of 61.0%). GA_Sharpe converged at an average of 64.271 steps, which means it calculated only $64.271 \times 25 = 1607$ sets of parameters among the 2^{12} sets (a reduction of 60.8%).

The results in Section 4.2 show that GA_Ret_SL converged at an average of 70.858 steps, which means it calculated only 1771 sets of parameters among the 2^{14} sets (a reduction of 89.2%). GA_Sharpe_SL converged at an average of 71.215 steps, which means it calculated only 1780 sets of parameters among the 2^{14} sets (a reduction of 89.1%).

The results in Section 4.2 show that GA_Ret_SLTP converged at an average of 73.813 steps, which means it calculated only 1845 sets of parameters among the 2^{16} sets (a reduction of 97.2%).

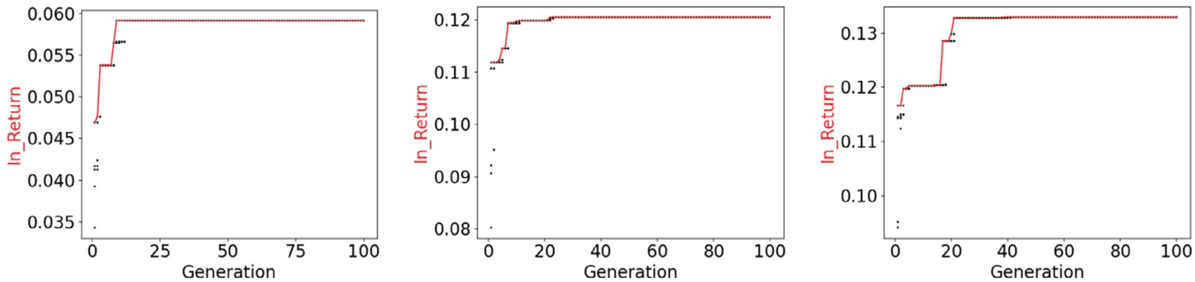

Fig. 10. The convergence of the GA_Ret, GA_Ret_SL, and GA_Ret_SLTP in Jan. 2007.

GA_Sharpe_SLTP converged at an average of 75.307 steps, which means it calculated only 1883 sets of parameters among the 2^{16} sets (a reduction of 97.1%).

Our experiment results show that computational overhead did not increase linearly with the size of the solution space. Compared to the generic ORB strategy, the GA_Ret and GA_Sharpe models reduced the computational overhead by approximately 60% (with improved profitability and stability), GA_Ret_SL and GA_Sharpe_SL models reduced the computational overhead by approximately 89% (with improved profitability and stability), and GA_Ret_SLTP and GA_Sharpe_SLTP models reduced the computational overhead by approximately 97% (with a decrease in performance). These experimental results demonstrate the efficacy of GA_Ret and GA_Sharpe with and without protective closing strategies in terms of time efficiency and profitability.

5. Conclusion

Any increase in trading frequency leads to an explosive growth in the quantity of technical data; therefore, it is necessary to develop efficient algorithms based on computational intelligence to deal with large datasets and high frequency trading. Opening range breakout (ORB) is a logical information-driven analysis strategy; however, it does not take into account all of the historical data or market characteristics. Furthermore, the fact that all positions are closed when the market closes can result in enormous losses when trading signals are unreliable or lack sufficient stability. Essentially, ORB lacks robustness against market uncertainties, such as information from multiple or contradictory sources.

In this study, we developed a GA-based framework (GAORB) to improve trading performance, while enhancing computational efficiency. We also developed two variants of the GAORB framework (GAORB_Ret and GAORB_Sharpe) to take into account a variety of selection criteria. Finally, we developed protective closing strategies using stop-loss and take-profit mechanisms to prevent unbearable losses and maximize profits.

In experiments, GA_Ret and GA_Sharpe enhanced annual returns and the Sharpe ratio by 1.667% (to 8.068% and 8.305%) and 0.365 (to 1.810 and 1.865) over the original strategy, while reducing computational overhead by 60%. GA_Ret_SL and GA_Sharpe _SL increased the size of the solution space by 4 times; however, the proposed GA reduced computational overhead by 89%, while enhancing stability and profitability with annual returns of 9.303% and Sharpe ratio of 2.495. In summary, threshold adjustments via optimization were shown to enhance profitability, and stop-loss mechanisms were shown to stabilize profit performance. Thus, we recommend GA_Ret_SL and GA_Sharpe_SL. In the future, the designed ORB model will be extended to adopt other evolutionary algorithms (e.g., PSO, ACO) to see whether the performance can be further improved. Moreover, the multi-objective models could be an alternative way that can be incorporated with the designed ORB model for further research and discussion.

CRediT authorship contribution statement

Mu-En Wu: Conceptualization, Methodology, Formal analysis. **Jia-Hao Syu:** Conceptualization, Methodology, Formal analysis, Experimental validation. **Jerry Chun-Wei Lin:** Formal review and editing. **Jan-Ming Ho:** Conceptualization, Methodology, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research is supported by Ministry of Science and Technology (MOST), Taiwan, and the project code is MOST 109-2221-E-027-106-.

References

[1] E. Chan, Quantitative Trading: How to Build Your Own Algorithmic Trading Business, Vol. 430, John Wiley & Sons, 2009.
[2] G. Vidyamurthy, Pairs Trading: Quantitative Methods and Analysis, Vol. 217, Wiley Fiance, 2004.
[3] Z. Qiang, Y. Shu-e, Noise trading, investor sentiment volatility, and stock returns, Syst. Eng.-Theory Pract. 29 (3) (2009) 40–47.
[4] T. Foucault, J. Hombert, I. Roşu, News trading and speed, J. Finance 71 (1) (2016) 335–382.
[5] T.C. Green, Economic news and the impact of trading on bond prices, J. Finance 59 (3) (2004) 1201–1233.
[6] K. Subramanyam, Financial Statement Analysis, Includes Index, 2009.
[7] J.S. Abarbanell, B.J. Bushee, Fundamental analysis, future earnings, and stock prices, J. Account. Res. 35 (1) (1997) 1–24.
[8] D.P. Brown, R.H. Jennings, On technical analysis, Rev. Financ. Stud. 2 (4) (1989) 527–551.
[9] J.J. Murphy, Technical Analysis of the Financial Markets: a Comprehensive Guide to Trading Methods and Applications, New York Institute of Finance, 1999.
[10] J. Brogaard, T. Hendershott, R. Riordan, High-frequency trading and price discovery, Rev. Financ. Stud. 27 (8) (2014) 2267–2306.
[11] S.H. Chen, P.P. Wang, Computational intelligence in economics and finance, in: Computational Intelligence in Economics and Finance, 2004, pp. 3–55.
[12] B. Fang, P. Zhang, Big data in finance, in: Big Data Concepts, Theories, and Applications, 2016, pp. 391–412.
[13] J. Hanousek, R. Podpiera, Information–driven trading at the prague stock exchange: Evidence from intra–day data, Economics Transit. 10 (3) (2002) 747–759.
[14] L.K. Chan, N. Jegadeesh, J. Lakonishok, Momentum strategies, J. Finance 51 (5) (1996) 1681–1713.
[15] U. Holmberg, C. Lönnbark, C. Lundström, Assessing the profitability of intraday opening range breakout strategies, Finance Res. Lett. 10 (1) (2013) 27–33.
[16] Y.C. Tsai, M.E. Wu, J.H. Syu, C.L. Lei, C.S. Wu, J.M. Ho, C.J. Wang, Assessing the profitability of timely opening range breakout on index futures markets, IEEE Access 7 (2019) 32061–32071.
[17] J.H. Syu, M.E. Wu, S.H. Lee, J.M. Ho, Modified ORB strategies with threshold adjusting on Taiwan futures market, in: IEEE Conference on Computational Intelligence for Financial Engineering & Economics, 2019, pp. 1–7.

[18] P. Zhang, S. Shu, M. Zhou, An online fault detection model and strategies based on SVM-grid in clouds, IEEE/CAA J. Autom. Sinica 5 (2) (2018) 445–456.
[19] P.J.M.v. Laarhoven, E.H.L. Aarts, Simulated annealing, in: Simulated Annealing: Theory and Applications, 1987.
[20] S. Ruder, An overview of gradient descent optimization algorithms, 2016, arXiv preprint arXiv:1609.04747.
[21] D. Whitley, A genetic algorithm tutorial, Stat. Comput. 4 (2) (1994) 65–85.
[22] D.B. Fogel, What is evolutionary computation? IEEE Spectr. 37 (2) (2000) 26–32.
[23] A.J. Keane, Genetic algorithm optimization of multi-peak problems: studies in convergence and robustness, Artif. Intell. Eng. 9 (2) (1995) 75–83.
[24] J.A. Endler, Natural selection in the wild, Princeton University Press, 1986.
[25] A. Warburton, Z.G. Zhang, A simple computational model for analyzing the properties of stop-loss, take-profit, and price breakout trading strategies, Comput. Oper. Res. 33 (1) (2006) 32–42.
[26] J.H. Holland, et al., Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, MIT press, 1992.
[27] A. Santiago, B. Dorronsoro, A.J. Nebro, J.J. Durillo, O. Castillo, H.J. Fraire, A novel multi-objective evolutionary algorithm with fuzzy logic based adaptive selection of operators: FAME, Inform. Sci. 471 (2019) 233–251.
[28] J.M. Smith, The Theory of Evolution, Cambridge University Press, 1993.
[29] O. Kramer, Genetic Algorithm Essentials, Vol. 679, 2017.
[30] J.E. Baker, Reducing bias and inefficiency in the selection algorithm, in: The International Conference on Genetic Algorithms and their Applications, 206, 1987, pp. 14–21.
[31] J.E. Baker, Adaptive selection methods for genetic algorithms, in: The International Conference on Genetic Algorithms, 1985, pp. 101–111.
[32] A. Lipowski, D. Lipowska, Roulette-wheel selection via stochastic acceptance, Physica A 391 (6) (2012) 2193–2196.
[33] J. Grefenstette, R. Gopal, B. Rosmaita, D. Van Gucht, Genetic algorithms for the traveling salesman problem, in: The International Conference on Genetic Algorithms, 1985, pp. 160–168.
[34] P. Larranaga, C.M.H. Kuijpers, R.H. Murga, I. Inza, S. Dizdarevic, Genetic algorithms for the travelling salesman problem: A review of representations and operators, Artif. Intell. Rev. 13 (2) (1999) 129–170.
[35] A. Ghosh, B. Nath, Multi-objective rule mining using genetic algorithms, Inform. Sci. 163 (1–3) (2004) 123–133.
[36] T. Mareda, L. Gaudard, F. Romerio, A parametric genetic algorithm approach to assess complementary options of large scale windsolar coupling, IEEE/CAA J. Autom. Sinica 4 (2) (2017) 260–272.
[37] Y. Hou, N. Wu, M. Zhou, Z. Li, Pareto-optimization for scheduling of crude oil operations in refinery via genetic algorithm, IEEE Trans. Syst. Man Cybern. Syst. 47 (3) (2015) 517–530.
[38] N. Metawa, M.K. Hassan, M. Elhoseny, Genetic algorithm based model for optimizing bank lending decisions, Expert Syst. Appl. 80 (2017) 75–82.
[39] L. Nunez Letamendia, Fitting the control parameters of a genetic algorithm: An application to technical trading systems design, European J. Oper. Res. 179 (3) (2007) 847–868.
[40] J. Díaz, M. Cortés, J. Hernández, Ó. Clavijo, C. Ardila, S. Cabrales, Index fund optimization using a hybrid model: genetic algorithm and mixed-integer nonlinear programming, Eng. Econ. 64 (3) (2019) 298–309.
[41] S. Srinivasan, T. Kamalakannan, Multi criteria decision making in financial risk management with a multi-objective genetic algorithm, Comput. Econ. 52 (2) (2018) 443–457.
[42] D.G. Uyemura, C.C. Kantor, J.M. Pettit, Eva® for banks: Value creation, risk management, and profitability measurement, J. Appl. Corp. Finance 9 (2) (1996) 94–109.
[43] B.M. Barber, T. Odean, Trading is hazardous to your wealth: The common stock investment performance of individual investors, J. Finance 55 (2) (2000) 773–806.
[44] A.W. Lo, The statistics of sharpe ratios, Financ. Anal. J. 58 (4) (2002) 36–52.
[45] A.C. Briza, P.C. Naval Jr, Stock trading system based on the multi-objective particle swarm optimization of technical indicators on end-of-day market data, Appl. Soft Comput. 11 (1) (2011) 1191–1201.
[46] G.N. Gregoriou, J.-P. Gueyie, Risk-adjusted performance of funds of hedge funds using a modified sharpe ratio, J. Wealth Manage. 6 (3) (2003) 77–83.
[47] D.H. Bailey, M. Lopez de Prado, The sharpe ratio efficient frontier, J. Risk 15 (2) (2012) 1–36.
[48] H.J. Solberg, The profit factor in fire insurance rates, J. Insurance (1957) 24–33.
[49] Z. Huang, F. Martin, Pairs trading strategies in a cointegration framework: back-tested on CFD and optimized by profit factor, Appl. Econ. 51 (22) (2019) 2436–2452.
[50] S. Lou, Grain contribution and profit deviation measure in Heilongjiang province based on factor analysis, in: The International Conference on Mechatronics, Computer and Education Informationization, pp. 345–349.
[51] L. Pospisil, J. Vecer, Portfolio sensitivity to changes in the maximum and the maximum drawdown, Quant. Finance 10 (6) (2010) 617–627.
[52] B.V. de Melo Mendes, R.C. Lavrado, Implementing and testing the maximum drawdown at risk, Finance Res. Lett. 22 (2017) 95–100.
[53] W.F. Sharpe, The sharpe ratio, J. portf. Manage. 21 (1) (1994) 49–58.
[54] M. Magdon-Ismail, A.F. Atiya, Maximum drawdown, Risk Mag. 17 (10) (2004) 99–102.
[55] B. Babcock, M. Datar, R. Motwani, Sampling from a moving window over streaming data, in: The Annual ACM-SIAM Symposium on Discrete Algorithms, 2002, pp. 633–634.
[56] P. Baldi, Computing with arrays of bell-shaped and sigmoid functions, in: Advances in Neural Information Processing Systems, 1991, pp. 735–742.