Electronic Commerce Research
https://doi.org/10.1007/s10660-021-09467-y

An intelligent trading mechanism based on the group trading strategy portfolio to reduce massive loss by the grouping genetic algorithm

Chun-Hao Chen[1] · Yu-Hsuan Chen[2] · Vicente Garcia Diaz[3] · Jerry Chun-Wei Lin[4]

Accepted: 17 February 2021

Abstract

It is always difficult and challenge to obtain suitable trading signals for the desired securities in financial markets. The popular way to deal with it is through the use of trading strategies (TSs) made up of technical or fundamental indicators. Due to the different properties of TSs, an algorithm was proposed to find trading signals by obtaining the group trading strategy portfolio (GTSP), which is composed of strategy groups that can be employed to generate various TS portfolios (TSP) instead of a single TS. The stop-loss and take-profit points (SLTP) are widely utilized by shareholders to avoid massive losses. However, the appropriate SLTP is hard to set by users. Therefore, in this paper, the algorithm, namely GTSP-SLTP algorithm, is proposed to not only obtain a reliable GTSP but also find appropriate SLTP using the grouping genetic algorithm. A chromosome is encoded by the generated SLTP and GTSP along with the weights for strategy groups that are the SLTP, grouping, weight, and strategy parts. To assess the goodness of a chromosome, the evaluation function that consists of the group balance, weight balance, risk factor, and profit factor, is employed. Genetic operators are then performed to produce new solutions for next population. The genetic process is performed iteratively until the stop conditions have achieved. Last but not the least, empirical experiments were conducted on three financial datasets with different trends and a case study is also given to reveal the effectiveness and robustness of the designed GTSP-SLTP algorithm.

Keywords Grouping problem · Grouping genetic algorithm · Portfolio optimization · Stop-loss · Take-profit · Trading strategy

This is a modified and expanded version of the paper " A sophisticated optimization algorithm for obtaining a group trading strategy portfolio and its stop-loss and take-profit points," presented at the IEEE International Conference on Systems, Man, and Cybernetics, 2018.

✉ Jerry Chun-Wei Lin
jerrylin@ieee.org

Extended author information available on the last page of the article

Published online: 10 March 2021

1 Introduction

Due to the complexity of the stock markets, portfolio optimization is always an attractive and challenging research field in financial markets. Various securities such as stocks [10, 23], options [15], or futures [35] can be used to form a portfolio. For users in the markets, the main purpose is to figure out a portfolio that can provide maximum profit and avoid massive loss because the markets are easily influenced by many and different factors, e.g., economic or political factors [24]. The Mean–Variance model is the well-known model for deriving an efficient frontier which refers to a set of portfolios [30]. It is difficult to obtain the efficient frontier, thus many algorithms are presented to allocate weights of a desired set of assets using evolutionary algorithms [2, 26, 28, 37].

When a portfolio is suggested, the next concern is when the securities should be purchased and sold and how capital should be allocated. Users may have several ways to identify them but no one can guarantee the found trading signals is proper and suitable, and the popular way to decide signals for buying and selling securities is based on trading strategies (TSs). Generally, TSs may be constructed by technical or fundamental indicators [9, 11, 24]. Since it is not an easy task to develop the effective TSs, many algorithms were respectively designed and implemented based on their specific strategies. For instance, some methods have been presented to search the TSs that can be used to determine buying and selling signals for assets particular in portfolio management [1]25. Some approaches were designed to obtain appropriate parameters of TSs, which may not always be given by experts or in advance [14]34.

Because a robust trading plan may consider a set of TSs together, optimization techniques have been designed to obtain the trading strategy portfolio (TSP) [5, 9]. To deliver more effective TSP, the optimization algorithm has been presented for finding a group trading strategy portfolio (GTSP) [4] using the grouping genetic algorithm (GGA). A GTSP considers a set of TS groups where a TS group contains many TSs with similar properties. For instance, assume that a GTSP consists of three TS groups where each group has three TSs, then the twenty-seven TSPs can be suggested to users. To put it another way, a GTSP provides users a more effective mechanism to make trading plans.

However, the overtrading problem is not considered in the previous work [4] since many trading signals could be identified using a given TS. While taking the trading cost into consideration, even return of every trading is positive, the cumulative return, however, may become negative. To handle this issue, the stop-loss and take-profit points (SLTP) are the common ways to be employed that can increase the return and to avoid massive loss.

To set appropriate SLTP for TS [4] and solve the limitation of the past works, in this paper, we propose an algorithm to obtain a GTSP and its SLTP using the GGA, namely the GTSP-SLTP algorithm. For a chromosome, it first randomly generates SLTP and forms the candidate TSs using the selected ten technical indicators. The generated SLTP and ranking functions are utilized to keep qualified TSs that are used to form TS groups and generate a possible GTSP. Then, the generated SLTP and GTSP along with the weights for strategy groups are encoded into a chromosome according to the encoding scheme which is represented by the SLTP as grouping, weight, and strategy parts. In the same way, the initial population can be initialized. To evaluate the fitness of every possible solution, the fitness function which composes of four factors that are the profit factor, the risk factor, the weight balance, and the group balance is utilized. The profit factor is calculated by the sum of returns of the TSPs with the SLTP can be generated from a chromosome. The risk factor is calculated by the maximum draw down of TSPs. The other two factors are used to measure the balance degree of TS groups of a GTSP in terms of numbers of strategies and weights in a chromosome. To maintain the diversity of chromosomes, the genetic operators are executed to produce new offspring. The evolution process continues until the termination criteria are reached. Empirical experiments were conducted on the three datasets with the uptrend, sideway trend and downtrend, and a case study is also given to show the merits of the GTSP-SLTP algorithm. In summary, the contributions of this work are listed as follows:

(1) *Providing an intelligent trading mechanism* The GTSP-SLTP algorithm can not only provide a useful mechanism for investors to make trading plans through the obtained GTSP, but also guide an effective way to keep profits and to limit losses within an acceptable range through the obtained SLTP.
(2) *Avoiding massive loss in bear market* Comparing the GTSP-SLTP algorithm with the existing approach [4] and the well-known strategy, the buy and hold strategy (BHS), experimental results on the three datasets indicate the GTSP-SLTP algorithm is effective in terms of return particularly in bear markets.
(3) *Generating appropriate trading signals for assets* For a case study, experiments were also made to show the merits of group stock portfolio (GSP) [6] with the GTSP-SLTP algorithm. The results reveal that the variance of return of GSP with GTSP-SLTP algorithm is smaller than that without it.

The remaining part of this paper is organized as follows. Relate work and background knowledge are reviewed in Sect. 2. Motivation and problem definition are stated in Sect. 3. Section 4 describes components of the GTSP-SLTP algorithm. The flowchart, pseudo code and an example are used to described the GTSP-SLTP algorithm in Sect. 5. Extensive experiments are discussed in Sect. 6. Finally, Sect. 7 provides conclusions and future work.

2 Literature reviews and background knowledge

Related studies and background knowledge are introduced in this section. The review of trading strategy optimization approaches is described in Sect. 2.1. The GGA and grouping problem are introduced in Sect. 2.2

2.1 Review of trading strategy optimization

Currently, many approaches have been proposed for the TS optimization, and they can be divided into two categories that are the TS optimization without and with SLTPs. For the TS optimization without SLTPs, many approaches have been proposed to solve the TS parameter optimization [14, 26, 34], incorporating TS in stock trading [1, 3, 20, 25, 33, 36], and the TSP optimization [4, 5, 32, 24].

In TS parameter optimization, Fu et al. proposed a genetic-based approach for determining the appropriate parameter settings of the selected technical indicators [14]. It generates TSs in accordance with the seven technical indicators. Then, the parameters of those TSs are encoded into a two-dimension array to represent a possible solution. The profit of a chromosome is used as a fitness function to identify appropriate parameters. Since optimization of TS parameters is time-consuming, Qin et al. presented a MapReduce-based algorithm to speed up the learning processing [34]. The presented approach consists of two MapReduce jobs. The first job is to generate the permutations of parameter combinations. For each combination, the second job is launched to calculate performance metrics of TSs. At last, the best parameter combination is determined according to the performance metrics. Lin et al. proposed a statistical learning method to find the most useful pair from multiple pair assets by combining both diversification and pair trading [29]. The results suggested the strategy can help investments be more diversified and profitable in stock trading.

To incorporate TS in stock trading, Chang et al. considered Markov decision process in GA to formulate TSs for stock markets [1], thus the trading signals are obtained by the Markov decision process. The GA is then employed to search the optimal stock selection strategy and capital allocation. Chien et al. proposed an GA-based approach to build an associative classifier. It can generate trading rules with the given numerical technical indicators [3]. In addition, Wu et al. proposed the adaptive stock trading strategies based on the deep reinforcement learning for trading. It first uses the gated recurrent unit to extract the features for making trading decisions. Two trading strategies with reinforcement learning methods are then presented as gated deep Q-learning trading strategy and gated deterministic policy gradient trading strategy to obtain the state-action table. Results showed that their approach can not only outperform the Turtle trading strategy but also has more stable returns [36]. Part et al. proposed an intelligent financial portfolio trading strategy using the deep Q-learning [33]. In their approach, the deep Q-learning is employed to train the intelligent agent and identify the optimal trading action. Ha et al. proposed an optimal intraday trading algorithm for reducing overall transaction costs when an online portfolio selection method rebalances the portfolio, and the results indicated the algorithm is significant to reduce the transaction costs when the liquidity is limited [20].

In TSP optimization, Chou et al. designed an algorithm to construct a rule-based dynamic trading system for stock [24]. In their approach, the technical indicators are used to generate TSs. An optimal combination of TSs is then obtained using the quantum-inspired Tabu search algorithm. The sliding window is also taken into consideration to avoid over-fitting and to achieve dynamic system. Chen et al. designed a combination genetic algorithm for building investment strategy portfolio [5]. It first uses ten technical indicators that every indicator has the selling and buying signals, and ten stocks to form a thousand security-rule pairs. A possible TSPs is then generated in terms of returns and encoded into a chromosome based on the top ten percent strategies. Return of a portfolio is employed as the evaluation function to score chromosomes. Nuij et al. designed a framework for automatic exploitation of news in stock trading strategies [32]. In that approach, events are first extracted from news. Considering the extracted events and technical indicators, the designed framework is used to find TSs using genetic programming, where the fitness of a trading strategy is evaluated by return calculation according to the given dataset. After several evolutions, they indicated that the news variable is often appeared in the optimized trading strategies, which means that the proposed framework is effective.

For the TS optimization with SLTPs, Kaminski et al. proposed an approach to analyze the stop-loss strategies [21]. Based on the three stop-loss cases, they investigate the results of the stop-loss policy. The cases are the regime-switching models, the mean reversion and momentum, and the random walk hypothesis. The analyzing results showed that the stopping premium which means the marginal impact of stop-loss rules on expected return of a given portfolio is always negative for the random walk hypothesis. However, in other two cases, stop-loss strategies can reach positive stopping premium. Lo et al. presented the closed-form expressions for the impact of stop-loss strategies on security returns that are serially correlated, regime switching, and subject to transaction costs [27]. They describe that tight stop-loss strategies could have worse return because of excessive trading costs when comparing to the BHS. Stop-loss outperformance is also possible for those assets that have high correlation in returns. Using GA, Leu et al. presented an algorithm for obtaining stock portfolio trading strategy with weighted fuzzy time series [25]. In first step, the stock portfolio is obtained using GA. The weighted fuzzy time series is then used to calculate the fitness value. The periodically checking and stop-loss point checking are used to decide trading signals of the stock portfolio.

2.2 The grouping problem and the grouping genetic algorithm

The main purpose of the grouping problem is to divide instances into a predefined number of groups by considering criteria that not only in groups but also between groups. Give a set of instance $INS = \{ins_1, ins_2, \ldots, ins_n\}$, the grouping problem is defined as following:

$$\cup G_i = INS \text{ and } G_i \cap G_j = \emptyset, i \neq j,$$

where G_i refers to the group G_i. Since it is time-consuming to obtain a grouping result with the criteria, the genetic algorithms (GA) which is one of evolutionary-based algorithms can be employed to handle it. Note that the merits of GA is that it can handles a variety of optimization problems effectively [16, 17, 19]. Based on the GA, the grouping genetic algorithm (GGA) was designed to solve various grouping problems and indicated that the performance of GGA is better than simple GA [12]13.

In the following, the details of GGA are stated. In encoding schema, the grouping and instance parts are used to present a solution. For example, a chromosome is given as follows:

AAABC: ABC.

In the chromosome, the string "AAABC" is the instance part before the colon, which refer to the five instances that are ins_1, ins_2, ins_3, ins_4 and ins_5. The string "ABC" is the grouping part after the colon, which indicates that instances in instance part can only belong to the three groups either A, B, or C. Therefore, the chromosome indicates that five instances are divided into three groups. The instances ins_1, ins_2, and ins_3 belong to group A. The instance o_4 belongs to group B, and the object o_5 belongs to group C.

As to genetic operators, the GGA has three operators that are crossover, mutation and inversion [13]. For the crossover operator, it switches groups in GGA instead of exchanging genes in GA. For mutation operator, it moves an instance from a group to another group. For the third operator, inversion, it changes the order of the groups in chromosome, and the goal of this operator is to increase the probability of crossover operator to get more diverse chromosomes. Literature also showed that the GGA can be efficiently used to handle stock portfolio optimization problem [6–8].

3 Motiviation and problem definition

In this section, the motivation is stated in Sect. 3.1, and the problem definition is given in Sect. 3.2, respectively.

3.1 Motivation

The motivation of this paper is to proposed an algorithm that can be utilized to obtain a GTSP and the suitable SLTPs. An example used to describe the motivation is shown in Fig. 1.

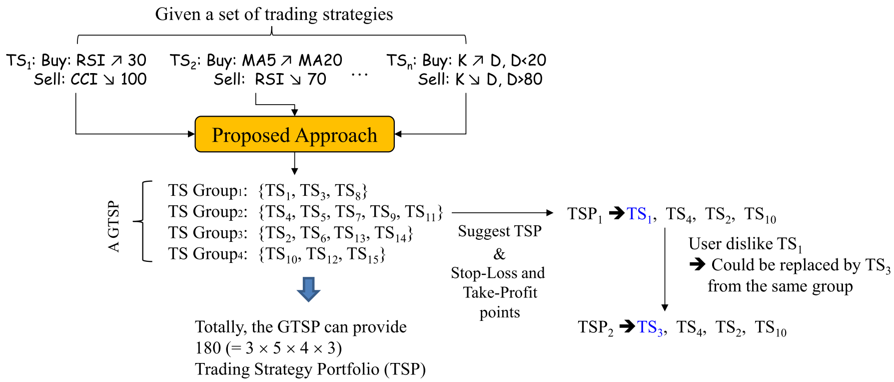

Fig. 1 An example to describe the motivation

From Fig. 1, a GTSP and the SLTPs are then considered in the designed approach to reduce loss and increase profit are derived based on the given set of trading strategies. In this example, the obtained GTSP contains four trading strategy groups. The first group has three trading strategies that are TS_1, TS_3 and TS_8. For other three groups, they have 5, 4 and 3 trading strategies. Hence, totally, 180 trading strategy portfolios can be provided by the GTSP to user. Assume the TSP_1 is suggested, when the user dislikes the trading strategy TS_1, it can be replaced by another trading strategy TS_3 from the same group. In addition, the suggested SLTPs can be utilized to prevent massive loss. To reach the mentioned goal, in other words, many factors should be considered, e.g., trading strategy groups, weight of groups, the SLTPs, and return, among others. To clarify that, the problem definitions are stated as follows.

3.2 Problem definition

In the following, six definitions are used for problem definition of the designed model.

Definition 1. Trading strategy (TS).trading strategy (TS) A TS consists of two rules that are buying and selling rules to generate buying and selling signals. Each rule cab be formed based on the technical indicators. Take the technical indicator, moving average (MA), as an example. A TS could be "Buying rule: When five-day MA crosses ten-day MA to the upside, a buying signal is generated; Selling rule: When five-day MA crosses ten-day MA to the downside, a selling signal is generated ".

Definition 2. Trading strategy portfolio (TSP) A TSP contains a set of TSs that can be expressed by $TSP = \{TS_1, TS_2, \ldots, TS_h\}$. For example, a TSP has two TSs possibly as {(TS_1: Buying rule: MA5 ↗ MA20; Selling rule: MA5 ↘ MA20), (TS_2: Buying rule: RSI ↗ 30; Selling rule: RSI ↘ 70)}.

Definition 3. Trading strategy group (TSG) A TSG is a set of TSs. The difference between TSP and TSG is that TSs in a TSG indicate that they have similar properties. For example, they could all suitable for trend trading or contrarian trading. A TSG is also denoted as $\{TS_1, TS_2, \ldots, TS_n\}$.

Definition 4. Group trading strategy portfolio (GTSP) A GTSP consists of K TSGs, and can be represented by $GTSP = \{TSG_1, TSG_2, \ldots, TSG_K\}$. Though a GTSP, $|TSG_1| \times |TSG_2| \times \ldots \times |TSG_K|$, TSPs can be generated. For example, given a GTSP that contains three groups, TSG_1, TSG_2 and TSG_3. Numbers of TSs in the three groups are 3, 3 and 4. Then, 32 ($= 3 \times 3 \times 4$) TSPs can be generated. To obtain a qualified GTSP is considered as a grouping problem. It means that criteria not only inside groups and but also between groups should be considered to evaluate quality of a solution.

Definition 5. Group trading strategy portfolio optimization (GTSPO) The aim of the GTSPO problem is to obtain a GTSP that can satisfy the predefined conditions inside groups and between groups using the heuristic algorithms, e.g., the conditions could be weights of groups, returns and risks of TSPs in a GTSP.

Definition 6. Group trading strategy portfolio with stop-loss and take-profit points optimization (GTSP-SLTPO) Based on the Definitions 4 and 5, the aim of the GTSP-SLTPO is to obtain not only a GTSP but also its SLTPs in accordance with the designed criteria using the heuristic algorithms to reach a robust performance.

Based on the abovementioned definitions, this paper proposes an optimization algorithm, namely GTSP-SLTP algorithm to solve the GTSP-SLTPO problem. Details of the proposed algorithm are stated in the following sections. Before that, the used abbreviations and expansions are summarized in Table 1.

4 Components of proposed approach

In this section, the chromosome representation is stated in Sect. 4.1. The fitness function and reproduction, as well as genetic operators are respectively described in Sects. 4.2 and 4.3.

4.1 Chromosome representation

While utilizing optimization approach for solving a problem, the design of the chromosome representation or encoding schema always be the first task that should be considered because it seriously influences the final results. In this paper, the aim is to obtain a GTSP and its SLTP. Thus, the SLTP, grouping, weight, and strategy parts are employed to represent a GTSP and its SLTP. The chromosome representation is shown in Fig. 2.

Figure 2 shows that the SLTP part is composed of bit strings to indicate the stop-loss and the take-profit thresholds. They are represented by n and b bits, respectively. In accordance with the SLTP represented in a chromosome, the trading signals, including selling and buying, of each TS can be located. When the

Table 1 The used abbreviations and expansion

Abbreviation	Expansion	Abbreviation	Expansion
TS	Trading Strategy	TSP	Trading Strategy Portfolio
TSG	Trading Strategy Group	GTSP	Group Trading Strategy Portfolio
SLTP	Stop-Loss and Take-Profit Points	GGA	Grouping Genetic Algorithm
SLP	Stop-Loss Point	BHS	Buy-and-Hold Strategy
TPP	Take-Profit Point	GSP	Group Stock Portfolio

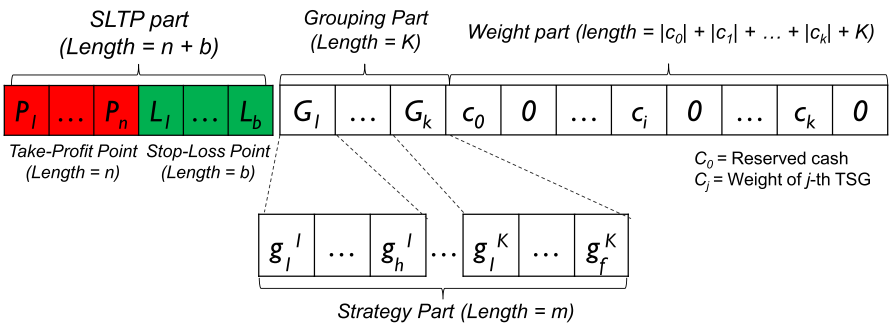

Fig. 2 Chromosome representation of a GTSP and its SLTP

return rate is larger than take-profit point (TPP*)* or less than stop-loss point (SLP), the asset will be sold. On the contrary, when the return rate is between TPP and SLP, the asset will be held. Note that the return rate is the different between selling and buying prices divides by the buying price. The grouping part indicates that the number of TSGs in a GTSP. The TS part reveals what TSs are included in the TSGs. As to the weight part, each c_j is used to indicate the allocated capital ratio of the *j*-th TSG and c_0 indicates the reserved capital, where a ’1′ string is utilized to represent a c_j, and the symbol ’0′ is used to separate the two adjacent weight strings. Using the encoding schema, chromosomes can be generated to formed the initial population for the evolution process.

An example is given as follows to show the encoded chromosome used in the designed algorithm. When the bound of the *TPP* and *SLP* are respectively set as 15% and – 15%, it indicates the ranges of them are in [0, 15%] and [0, – 15%]. Assume a TPP and SLP are represented by four bits, number of TSs is fifteen and the number of groups is four, the chromosome can be encoded in Fig. 3.

Figure 3 shows that the TPP is 8% and SLP is – 3% according to the bit strings "0100" and "0011". There are four TSGs that are G_1, G_2, G_3 and G_4. The group G_1 has five trading strategies, TS_4, TS_5, TS_6, TS_7 and TS_{14}. Since number of ’1′ in c_0 is 1 and total number of ’1′ is five in the weight part, it indicates the reserved capital is 20% (= 1/5). In the same way, we can observe that number of ’1′ in c_1 to c_4 are 0, 3, 1 and 0. The weights for G_1 to G_4 are 0%, 60%, 20% and 0%. Utilizing the chromosome, 10 (= 2 × 5) TSPs can be generated and suggested to users due to weights of G_1 and G_4 are zero.

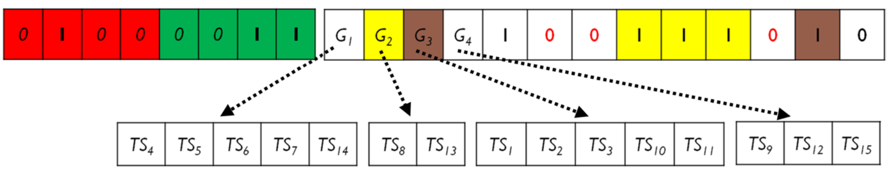

Fig. 3 An example of chromosome representation

To describe how to calculate the return of a TS using the encoded SLTP, trading signals generated by a TS and stock price series in Table 2 with the 8% and -3% as SLTP are employed to explain the process. Note that ’1′ and ’0′ respectively represent the buying and selling signals.

Table 2 shows that the first buying and selling signals appear on 2/5 and 2/10. Return rate for purchasing the stock is 6% (=(217–203)/203). In this case, the return rate is smaller than 8%, the asset will be held waiting next selling signal. Since the next selling signal appears on 2/12, the return rate is calculated as 11% (=(224.5–203) / 203). In this situation, the asset will be sold because the return rate is larger than the TPP which is 8%.

4.2 Fitness function and reproduction

It is a critical task to obtain a good GTSP and its SLTP by designing a proper fitness function. In other words, many factors should be considered in the design process of a fitness function. To identify a qualified GTSPs and is SLTP, two types of criteria should be considered. The first type is used to evaluate the profitable ability of a GTSP, and the second type is used to evaluate the structure of TSGs. The two types of factors are stated as follows: (1) For the profitable ability of a GTSP, it consists of the two sub-factors that are the return and risk should be maximized and minimized, and are used as the part of evaluation criteria; (2) For the structure of TSGs, the group balance and weight balance are taken into consideration for evaluating the balance degree of TSGs in terms of number of TSs and allocated capital. As a result, the fitness function which is composed of four factors to evaluate a chromosome is given in Formula (1).

$$f(C_q) = P\mathrm{Return}(C_q) * P\mathrm{Risk}(C_q) * GB(C_q)^{\alpha} * WB(C_q), \tag{1}$$

where $PReturn(C_q)$ and $PRisk(C_q)$ are the return and risk of a GTSP, $GB(C_q)$ and $WB(C_q)$ are the group balance and weight balance of TSGs, and α is a parameter to indicate the impact of group balance. The four criteria are stated as follows. The return of a GTSP in a chromosome is shown in Formula (2).

$$\mathrm{PReturn}(C_q) = \frac{\sum_{j=1}^{n\mathrm{TSP}} \mathrm{return}(\mathrm{TSP}_j)}{n\mathrm{TSP}}, \tag{2}$$

where $return(TSP_j)$ is the return of j-th TSP, and $nTSP$ is the number of TSPs generated from C_q. Note that the higher return of a GTSP is, the better GTSP is obtained. The $return(TSP_j)$ is stated in Formula (3).

$$\mathrm{Return}\ (\mathrm{TSP}_j) = \sum_{i=1}^{K} avgRRate(\mathrm{TS}_i^j) * weight_i * \mathrm{allocated\ Cap}, \tag{3}$$

where $avgRRate(TS_i^j)$ and $weight_i$ are the average return rate of the TS in group G_i of TSP_j and the weight of G_i, and $allocatedCap$ means the allocated capital of the TS for trading. The $avgRRate(TS_i{}^j)$ is stated in Formula (4).

Table 2 The trading signals and stock price series

Date	2/5	2/6	2/7	2/10	2/11	2/12	2/13	2/14	2/17	2/18	2/19	2/20	2/21	2/24	2/25
Open Price	191	203	213.5	221	217	221	224.5	222.5	216	218	220	216.5	218	218.5	219.5
Trading Signal	1			0		0	1				0			1	

$$\text{avgR Rate}(\text{TS}_i^j) = \frac{\sum_{h=1}^{frequent_i} return\ Rate(TS_{ih}^j)}{frequency_i} \tag{4}$$

where $frequency_i$ is the number of transactions during the given trading period, and $RRate(TS_{ih}{}^{j})$ is defined in Formula (5).

$$RRate(TS_{ih}^j) = \frac{\text{sell Price}_h - \text{buy Price}_h}{\text{buy Price}_h} \tag{5}$$

where the $sellPrice_h$ and $buyPrice_h$ are selling and buying prices of the h-th transaction using the i-th TS in TSP_j. Note that the $buyPrice_h$ is determined by the trading signal generated using the trading strategy, and the $sellPrice_h$ is determined by the SLTP part in the chromosome. For example, let the bit strings of the SLTP part are "0100" and "0100", the TPP and SLP are 8% and -8%. Then, the TPP and SLP will be used to determine selling price. The $PRisk(C_q)$ is defined in Formula (6).

$$\text{PRisk}(\text{C}_\text{q}) = \frac{\sum_{j=1}^{\text{nTSP}} \text{risk}(\text{TSP}_j)}{n\text{TSP}}, \tag{6}$$

where $risk(TSP_j)$ and $nTSP$ are the risk of a TSP and number of TSPs generated from C_q. The $risk(TSP_j)$ is shown in Formula (7).

$$\text{Risk}(\text{TSP}_\text{j}) = \min\left(\text{MDD}(TS_1^\text{j}) \ldots, \text{MDD}\left(TS_k^\text{j}\right)\right), \tag{7}$$

where $MDD(TS_i^j)$ and K are the maximum draw down (MDD) of the i-th TS of TSP_j and number of groups. It indicates that the risk of a TSP is calculated by the minimum MDD of the TSs in the portfolio. Note that the MDD of a TS is normalized to 0 to 1. The $MDD(TS_i^j)$ is defined in Formula (8).

$$\text{MDD}(TS_i^j) = \min\left(R\ \text{Rate}(TS_{i1}^j) \ldots, R\ \text{Rate}\left(TS_{ifrequency_i}^j\right)\right) \tag{8}$$

where $RRate(TS_{ih}{}^{j})$ is given in in Formula (5), and $frequency_i$ is number of transactions using the i-th TS in TSP_j during the given trading period. The group balance of TSGs in a GTSP is given in Formula (9).

$$GB(C_q) = \sum_{i=1}^{K} -\frac{|G_i|}{N} \log \frac{|G_i|}{N} \tag{9}$$

where $|G_i|$ and N are number of TSs in G_i and the number of the given TSs. The main purpose of group balance is to make the number of TSs in groups as the same as possible. The fourth factor is shown in Formula (10).

$$WB(C_q) = \sum_{i=1}^{K+1} -\frac{|c_i|}{TL} \log \frac{|c_i|}{TL} \tag{10}$$

where $|c_i|$ and TL are the length of the string c_i and the total length of all strings c_i, $0 \le i \le K$. The main purpose of weight balance is utilized for avoiding allocated capital at certain groups. Using the evaluation function, the fitness value of a possible solution can be calculated. According to the selection strategy, the next population will be generated, e.g., the elitist selection, the roulette wheel selection.

4.3 Genetic operators

This section describes the three genetic operators used in the GTSP-SLTP algorithm. The first operator is crossover which is executed on only the SLTP and weight parts. The one-point crossover and two-point crossover operators are applied on the SLTP and the weight parts to generate new offspring, respectively. Because applying crossover on the weight part may disrupt the number of "0" and "1" in a chromosome, the suitable arrangement should be done to correct them.

As to mutation operators, they are executed on the SLTP, TS and weight parts. To perform mutation on the SLTP part, one gene is randomly chosen for mutation. If the gene value is 0, it will be changed to 1; otherwise, it will be changed to 0. To perform mutation on the TS part, it will select and move a TS from a TSG to another TSG. For mutation on the weight part, a "0″ and a"1″ genes will be selected for exchanging. The third operator, the inversion, is performed on the grouping part. Due to the aim of this operator is to increase the diversity of a chromosome when executing crossover operator, it only exchanges the order of two random selected TSGs.

5 Proposed method

This section describes the proposed algorithm, namely the GTSP-SLTP algorithm, to obtain a GTSP and its SLTP using the GGA. In the following, the flowchart of the proposed approach is illustrated in Sect. 5.1. The pseudo code of the GTSP-SLTP algorithm is given in Sect. 5.2 and followed by an example in Sects. 5.3.

5.1 Flowchart of proposed approach

Based on the mentioned definitions above, in this paper, we propose an optimization algorithm, namely GTSP-SLTP algorithm, to solve the GTSP-SLTPO problem. The flowchart of the GTSP-SLTP algorithm is illustrated in Fig. 4.

Figure 4 shows that in accordance with the selected technical indicators and the stock price series, the initial population is first generated. Every chromosome means a potential GTSP and its SLTPs. The four parts of a chromosome are generated as follows. In the SLTP part, two randomly generated bit strings are used to represent the stop-loss threshold and take-profit threshold. Then, K groups are initialized for the grouping part. The TSs in groups are generated using m candidate TSs that are generated by the candidate TS generation procedure which will be described in Fig. 5. The weight part is represented by a randomly generated bit string. In the chromosome evaluation, four factors that are the portfolio return, the risk of portfolio, the group balance and the weight balance of groups are employed to calculate

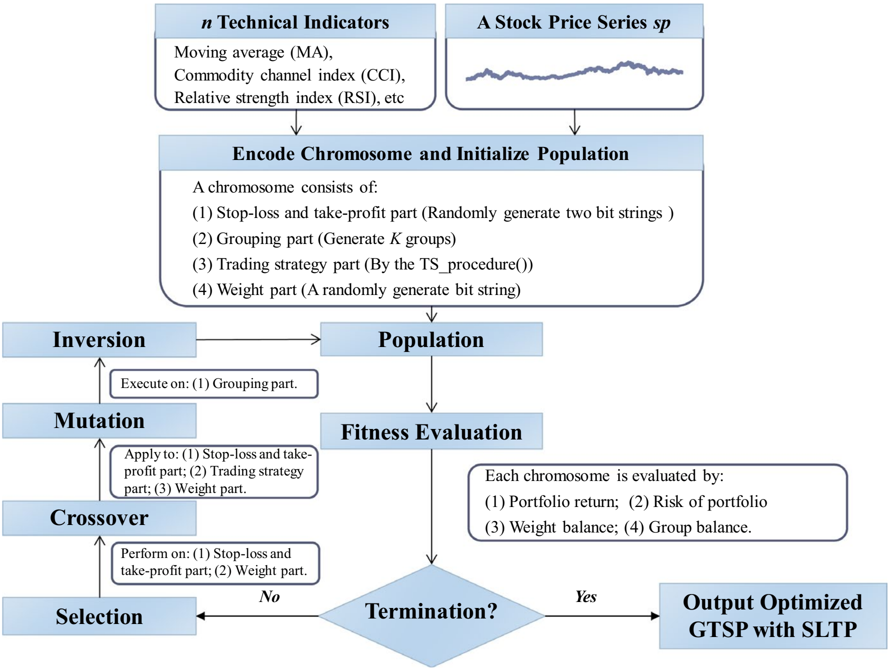

Fig. 4 Flowchart of the GTSP-SLTP algorithm

the fitness value of a chromosome. The four factors can be calculated by Formulas (2), (6), (9) and (10). Three genetic operators, including crossover, mutation and inversion, are executed on the population to generate new chromosomes. Finally, the obtained GTSP with its SLTPs is provided to investors. Below, Fig. 5 shows how the candidate TSs generation procedure work.

In the Fig. 5, it shows the processed m TSs are formed based on the given the stock prices series, technical indicators, and the SLTP part in a chromosome. The process consists of four phases: (1) It first forms candidate TSs by the selected technical indicators; (2) Then, according to the given stock price series and candidate TSs, the selling and buying signals are identified. Using the SLTP part, the generated trading signals will be relocated. Take the take-profit point 5% as an example. Although the TS generates a selling signal on 2014/2/19, the stock will still be held because the cumulative return is 2% which is smaller than the threshold. Thus, that selling signal is removed. Since the next selling signal will be generated on 2014/03/11 and its cumulative return is 8%, the stock will be sold and the new selling signal is added; (3) After relocating the trading signals, the ranking functions that are the average return, trading frequency and maximum draw down (MDD) are used to calculate scores of the candidate TSs. When using the trading strategy for trading, the MDD can be used to evaluate its risk degree. Given a set of transactions with returns, the MDD means the one causes the highest loss. Hence, if the MDD value is larger than 0, it indicates that the used trading strategy is better than that smaller than 0. Hence, in the candidate TSs generation procedure, the MDD is

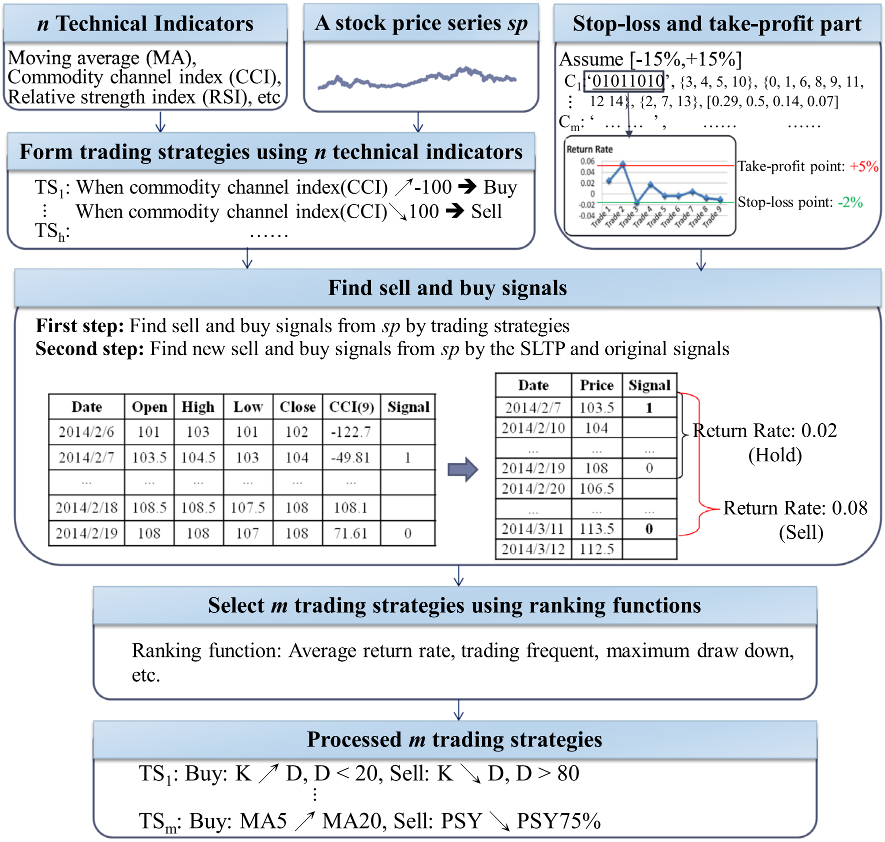

Fig. 5 The flowchart of candidate TSs generation procedure

employed as a ranking function to determine the m trading strategies; (4) Finally, using the preferred ranking function selection strategy, m TSs are selected to form the trading strategy part.

5.2 The pseudo code of the GTSP-SLTP algorithm

To state the proposed approach clearly in this section, the pseudo code of the GTSO-SLTP algorithm is given in Fig. 6.

Figure 6 shows the process of the GTSP-SLTP algorithm to obtain a GTSP and its SLTP using the GGA in accordance with the given stock price series *sp* and the technical indicators *nTech*. From lines 2 to 9, the initial population is first generated. From lines 11 to 18, based on the designed fitness function, every chromosome is evaluated by return and risk of portfolio, and group and weight balances. From lines 19 to 23, the genetic operators such as the crossover, mutation, inversion and selection operators, are utilized to generate new chromosomes. Finally, while reaching the predefined number of iterations, the chromosome with the highest fitness value

Proposed GTSP-SLTP Algorithm

Input:

A given stock price series *sp* and the selected technical indicators *nTech*.

Parameters:

Number of TSGs *K*, number of TSs *m*, population size *pSize*, crossover, mutation and inversion rates, p_c, p_m and p_i, number of iterations *numIter*, number of bits used to represent the weights *numweightBits*, predefined bounds of TPP *TPPBound* and SLP *SLPBound*, number of bits to represent TPP and SLP, *n* and *b*.

Output:

The optimized GTSP with its SLTPs.

Procedure: GTSP-SLTP-GGAOptimization

```
1.  GTSP_SLTP_GGA_Optimization (){
2.  population = { ϕ };
3.  FOR q = 0 to pSize DO
4.    sLATPPart = generateStopLossAndTakeProfitPointPart(TPP, SLP, n, b);
5.    tsPart = TS_Procedure(sp, nTech, m, sLATPPart);
6.    gPart = generateGroupingPart(K);
7.    wPart = generateWeightPartRandomly(K, weightBits);
8.    population = population ∪ mergedParts(sLATPPart, gPart, tsPart, wPart);
9.  END q FOR LOOP
10. FOR i = 0 to numIter DO
11.     FOR q = 0 to pSize DO
12.         gtspProfit ← PReturn(C_q, sp);
13.         gtspRisk ← PRisk(C_q, sp);
14.         gtspGB ← GB(C_q, α);
15.         gtspWB ← WB(C_q);
16.         fitValue ←gtspProfit * gtspRisk *gtspGB^α*gtspWB;
17.         population ← updatePopulation(C_q, fitValue);
18.     END q FOR LOOP
19.     tempPopu ← selectOperator(pSize, population);
20.     tempPopu ← crossOperator(p_c, tempPopu);
21.     tempPopu ← mutaOperator(p_m, tempPopu);
22.     tempPopu ← inverOperator(p_i, tempPopu);
23.     population ←tempPopu;
24. END i FOR LOOP
25. Output GTSPwithSLTP ← getBestChro(population);
26. }
```

Fig. 6 The pseudo code of the GTSP-SLTP Algorithm

will be produced as the optimized GTSP and the SLTP at line 25. The pseudo code of the TS procedure is given in Fig. 7.

Figure 7 indicates how the *m* TSs to be generated based on the selected technical indicators, stock price series and SLTP part of a chromosome. Firstly, the combinations of the selected technical indicators are generated to from the candidate TSs at

```
The TS Procedure
Input: A given stock price series sp, the selected technical indicators nTech and
SLTP part of a chromosome sLATPPart.
Parameter: Number of desired TSs m
Output: The top m TS set top_M_TS
Procedure: TS_Procedure()
1.  TS_Procedure(){
2.  candidateTS = generateTSCombi(nTech);
3.  signals = generateTradingSignal(sp, candidateTS, sLATPPart);
4.      FOR i = 0 to | candidateTS | DO
5.          avgReturn ← getAvgReturn(TS_i, signals);
6.          numFrequency ← getTradingFrequency(TS_i, signals);
7.          mdd ← getMaxDrawDown(TS_i, signals);
8.          scoredTS = aggregateFunction(avgreturn, numFrequency, mdd);
9.      END i FOR LOOP
10.     top_M_TS = getTop_M_TS(scoredTS, m);
11. Output top_M_TS;
}
```

Fig. 7 Pseudo code of TS procedure

line 2. Then, according to the candidate TSs and SLTPs, the trading signals can be identified from the given stock price series at line 3. Then, the average return, trading frequency and maximum draw down are used as the ranking functions to score TSs at lines 4 to 9. At last, the *m* TSs are selected based on the scores at lines 10 to 11.

5.3 An example

To illustrate the GTSP-SLTP algorithm, an example is provided in this section with eight steps.

STEP 1 Assume numbers of bits to represent TPP and SLP, population size, number of TSGs and number of TSs are 4, 10, 3 and 15, the initial population are generated by following sub-steps:

Sub-step 1.1 Eight bits are generated to represent SLTP part for each chromosome randomly. For instance, let the SLTP part of a chromosome C_1 is generated as "11,110,111". The first four bits mean the value of TPP and the followed four bits represent the value of SLP. Assume that the bounds of the TPP and SLP are 15% and -15%, according to the string "11,110,111", it means that the values of *TPP* and *SLP* are 15% (=0.15 / 15 × 15) and -7% (=0.15 / 15 × 7), respectively.

Sub-step 1.2 For each chromosome, the TS procedure is employed to generate 15 TSs according to its SLTP part, the stock price series and technical indicators. In this example, assume that the generated fifteen TSs of C_1 and their related data are given in Table 3.

In Table 3, the first attribute is the average return rate which is calculated using Formula (4). During the given trading period, the second attribute (*MDD*) represents the maximum loss of a TS and it can be calculated by the Formula (8).

Sub-step 1.3 Because the number of groups is three, the fifteen TSs are randomly divided into three groups to generate the grouping and strategy parts. For instance, the grouping and strategy parts of C_1 could be [G_1: {6, 8, 9, 12}, G_2: {4, 11}, G_3: {0, 1, 2, 3, 5, 7, 10, 13, 14}].

Sub-step 1.4 Assume that number of bits used to represent the weight part is 100, a weight part of C_1 could be generated as [1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1]. The length is 103 ($=100+3$) since the number of groups is 3. In other words, it indicates that weights of reserved capital and three groups are 0.13, 0.12, 0.72, 0.03. Note that for the sake of being concise, we will directly use real numbers transformed from the bit string in the following steps. Combining the four parts, the chromosome C_1 is represented as C_1: "11,110,111", [{6, 8, 9, 12}, {4, 11}, {0, 1, 2, 3, 5, 7, 10, 13, 14}], 0.13, 0.12, 0.72, 0.03.

Sub-step 1.5 Repeating the sub-steps 1.1 to 1.5 and since the population size is set as 10, the initial population are generated and shown as follows:

C_1: "11110111", [{6, 8, 9, 12}, {4, 11}, {0, 1, 2, 3, 5, 7, 10, 13, 14}], 0.13, 0.12, 0.72, 0.03;

Table 3 Generated TSs and their related data

TS_{id}	Avg. return rate	MDD
TS_0	0.281	0.281
TS_1	0.001	−0.108
TS_2	−0.027	−0.108
TS_3	−0.030	−0.248
TS_4	−0.048	−0.216
TS_5	−0.050	−0.232
TS_6	−0.052	−0.220
TS_7	−0.053	−0.226
TS_8	−0.055	−0.197
TS_9	−0.068	−0.307
TS_{10}	−0.072	−0.075
TS_{11}	−0.073	−0.073
TS_{12}	−0.074	−0.075
TS_{13}	−0.074	−0.075
TS_{14}	−0.081	−0.095

C_2 : "11100000", [{2, 4, 11, 13}, {1, 3, 8, 9, 10, 12, 14}, {0, 5, 6, 7}], 0.11, 0.15, 0.59, 0.15;
C_3: "11110111", [{4, 9, 12}, {0, 1, 3, 7, 11}, {2, 5, 6, 8, 10, 13, 14}], 0.43, 0.3, 0.21, 0.06;
C_4: "11010100", [{0, 2, 5, 6, 9, 10}, {3, 12, 13, 14}, {1, 4, 7, 8, 11}], 0.04, 0.19, 0.55, 0.22;
C_5: "10001011", [{2, 3, 11, 12, 13, 14}, {7, 8}, {0, 1, 4, 5, 6, 9, 10}], 0.12, 0.0, 0.65, 0.23;
C_6: "01011101", [{1, 3, 4, 9, 11, 12}, {0, 8, 10, 13}, {2, 5, 6, 7, 14}], 0.32, 0.56, 0.07, 0.05;
C_7: "01000001", [{0, 4, 7, 9, 11, 13}, {1, 5, 14}, {2, 3, 6, 8, 10, 12}], 0.0, 0.41, 0.33, 0.26;
C_8: "01110101", [{0, 2, 10, 11, 14}, {8, 9}, {1, 3, 4, 5, 6, 7, 12, 13}], 0.2, 0.22, 0.31, 0.27;
C_9: "01100100", [{7, 8, 11, 12, 14}, {0, 1, 2, 5, 6, 9, 13}, {3, 4, 10}], 0.34, 0.14, 0.13, 0.39;
C_{10}: "10111010", [{1, 7, 8, 12}, {0, 2, 3, 4, 10, 13, 14}, {5, 6, 9, 11}], 0.07, 0.08, 0.52, 0.33.

STEP 2 Every chromosome is evaluated by the designed fitness function via the following sub-steps:

Sub-step 2.1 The TSPs are first generated. Take chromosome C_1 as an example. In accordance with the grouping part [G_1: {6, 8, 9, 12}, G_2: {4, 11}, G_3: {0, 1, 2, 3, 5, 7, 10, 13, 14}], it generates 72 (=4×2×9) possible TSPs. All of them are collected in a set $tspSet = \{tsp_1, tsp_2, \ldots, tsp_{72}\} = \{\{6, 4, 0\}, \{6, 4, 1\}, \{6, 4, 2\}, \ldots, \{12, 11, 14\}\}$.

Sub-step 2.2 Using the following substeps, the returns of a chromosome is calculated.

Sub-step 2.2.1 Return of every TSP in the set *tspSet* is calculated. Take tsp_1: {6, 4, 0} of the chromosome C_1 as an example. In accordance with the weight part: [0.13, 0.12, 0.72, 0.03], let the investment capital is 100,000, the return of tsp_1 is -3237 (=[-0.052×(100,000×0.12)+-0.048×(100,000×0.72)+0.281×(100,000×0.03)]). In the same way, returns of remaining TSPs can be calculated.

Sub-step 2.2.2 After the previous subsetp, average return of TSPs is set as the return of a chromosome by Formula (2). The results of the ten chromosomes are given in Table 4.

Table 4 Portfolio returns of the ten chromosomes

C_q	$PReturn(C_q)$	C_q	$PReturn\ (C_q)$
C_1	−5161.653	C_6	−730.096
C_2	−147.480	C_7	1623.727
C_3	−1732.466	C_8	25.169
C_4	−2676.077	C_9	−222.141
C_5	−1297.195	C_{10}	−1758.062

Sub-step 2.3 Risk of a chromosome is then evaluated as follows:

Sub-step 2.3.1 The MDD of every TS is normalized firstly and the results are shown in Table 5.

Sub-step 2.3.2 The risk of each TSP tsp_j in the set *tspSet* is calculated. Take tsp_1: {6, 4, 0} as an example. The risk of tsp_1 is 0.148 (= min(0.148, 0.155, 1)) by Formula (7). After risk values of other TSPs are calculated, the risk of a chromosome can be set according to Formula (6). The results of all chromosomes are given in Table 6.

Sub-step 2.4 Based on the grouping part, the group balance of every chromosome is evaluated. Since the grouping part of C_1 is [G_1: {6, 8, 9, 12}, G_2: {4, 11}, G_3: {0, 1, 2, 3, 5, 7, 10, 13, 14}], the $GB(C_1)$ is calculated as 0.860. In the same way, the group balance scores of the ten chromosomes are given in Table 7.

Sub-step 2.5 Based on the weight part of a chromosome, the weight balance score is calculated. Take C_1 as an example. Since the weight part of C_1 is [0.13,

Table 5 Normalized MDD for every TS

TS_{id}	MDD	$Normalized\ MDD$	TS_{id}	MDD	$Normalized\ MDD$
TS_0	0.281	1	TS_8	−0.197	0.188
TS_1	−0.108	0.338	TS_9	−0.307	0
TS_2	−0.108	0.338	TS_{10}	−0.075	0.395
TS_3	−0.248	0.100	TS_{11}	−0.073	0.398
TS_4	−0.216	0.155	TS_{12}	−0.075	0.395
TS_5	−0.232	0.128	TS_{13}	−0.075	0.395
TS_6	−0.220	0.148	TS_{14}	−0.095	0.361
TS_7	−0.226	0.138			

Table 6 Risk of ten chromosomes

C_q	$PRisk(C_q)$	C_q	$PRisk(C_q)$
C_1	0.127	C_6	0.077
C_2	0.141	C_7	0.177
C_3	0.116	C_8	0.112
C_4	0.071	C_9	0.113
C_5	0.078	C_{10}	0.093

Table 7 Group balances of the ten chromosomes

C_q	$GB(C_q)$	C_q	$GB(C_q)$
C_1	0.860	C_6	1.178
C_2	1.125	C_7	1.113
C_3	1.089	C_8	0.941
C_4	1.178	C_9	1.089
C_5	0.982	C_{10}	1.125

Table 8 Weight balance of ten chromosomes

C_q	$WB(C_q)$	C_q	$WB(C_q)$
C_1	0.742	C_6	1.051
C_2	1.262	C_7	1.271
C_3	1.490	C_8	1.881
C_4	1.224	C_9	1.624
C_5	0.844	C_{10}	1.197

Table 9 Fitness values of ten chromosomes

C_q	$f(C_q)$	C_q	$f(C_q)$
C_1	−359.744	C_6	−81.991
C_2	−33.214	C_7	452.504
C_3	−355.112	C_8	4.695
C_4	−322.722	C_9	−48.345
C_5	−82.3503	C_{10}	−247.694

0.12, 0.72, 0.03] using Formula (10), the weight balance of chromosome C_1 is 0.742. The results of ten chromosomes are given in Table 8.

Sub-step 2.6 Let α is 2, the fitness value of C_1 can be set at -359.744 ($=-5161.653\times0.127\times0.860^2\times0.742$) according to Formula (1). The fitness values of the ten chromosomes are shown in Table 9.

Step 3 To generate next population, the elitist selection strategy is utilized for reproduction in this example.

Step 4 The crossover operator is executed to generate new chromosomes. Take chromosomes C_6 and C_9 as an example. Because the SLTP parts of C_6 and C_9 are "01011101" and "01100100", assume that the cutoff point is 3, after crossover, the SLTP parts of the new chromosomes are "01000100" and "01111101". For crossover on the weight part, take chromosomes C_3 and C_7 as an example, their weight parts are shown as follows:

[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1]

and

[0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1].

Assume that the cutoff points are respectively set as 3 and 74, the newly formed weight parts after transforming to real numbers are C_3': [0.42, 0.31, 0.21, 0.06] and C_7': [0.0, 0.42, 0.32, 0.26].

Step 5 The mutation operator is executed on the population to generate new chromosomes. Take chromosome C_3 as an example. For mutation on the SLTP part, let the mutation point is three, the SLTP part of C_3 changes from "11110111" to "11010111". For mutation on the TS part, assume the TS_4 in G_1 is selected and moved to G_2, the TS part of C_3 changes from [G_1: {**4**, 9, 12}, G_2: {0, 1, 3, 7, 11}, G_3: {2, 5, 6, 8, 10, 13, 14}] to [G_1: {9, 12}, G_2: {0, 1, 3, **4**, 7, 11}, G_3: {2, 5, 6, 8, 10, 13, 14}]. For mutation on the weight part, assume two genes, the 5-th and 44-th genes, are picked to exchange, the weight part of C_3 changes from [0.43, 0.3, 0.21, 0.06] to [**0.04**, **0.69**, 0.21, 0.06].

Step 6 The inversion operator is performed. Take chromosome C_5 as an example. Because the grouping part of C_5 is [G_1: {2, 3, 11, 12, 13, 14}, $\boldsymbol{G_2}$**: {7, 8}, G**$_\mathbf{3}$**: {0, 1, 4, 5, 6, 9, 10}**], let G_2 and G_3 are selected for exchanging, it changes to [G_1: {2, 3, 11, 12, 13, 14}, $\boldsymbol{G_2}$**: {0, 1, 4, 5, 6, 9, 10}, G**$_\mathbf{3}$**: {7, 8}**].

Step 7 If the termination condition is not achieved, go to Step 2 to perform the designed progress iteratively; otherwise, go to the next step.

Step 8 The chromosome with the highest fitness value is prodcued as the optimized GTSP and its SLTP. In this example, after 100 iterations, the final best chromosome C_{best} is ["10000100", G_1: {1, 3, 8, 9, 10, 12, 14}, G_2: {2, 4, 11, 13}, G_3: {0, 5, 6, 7}, 0.16, 0.15, 0.16, 0.53]. The C_{best} indicates that the fifteen TSs are divided into three groups. The TPP is 8% and the SLP is −4%. Group G_1 contains TS_1, TS_3, TS_8, TS_9, TS_{10}, TS_{12} and TS_{14}. Group G_2 composes of TS_2, TS_4, TS_{11} and TS_{13}. Group G_3 has TS_0, TS_5, TS_6, and TS_7. The weight part represents that allocated capital for groups. Based on the obtained GTSP, there are 112 TSPs (=7×4×4) can be provided to users making investment plans.

6 Experimental evaluations

In this section, three financial datasets with different trends are utilized to evaluate the effectiveness of the proposed GTSP-SLTP algorithm. Related parameters are given in Table 10.

In the following, the dataset descriptions and the experimental evaluations on the datasets are discussed in Sects. 6.1 and 6.2. Then, applying the GTSP-SLTP algorithm on the group stock portfolio (GSP) [6] to determine the trading signals to demonstrate the advantages of the GTSP-SLTP algorithm is given in Sect. 6.3. In other words, by using the GTSP-SLTP algorithm, we would like to evaluate whether the more appropriate trading signals generated by various trading strategies can be employed for reaching a better trading performance than the BHS.

Table 10 Parameter settings

Parameter	Value	Parameter	Value
Number of TSs	15	#bits of TPP	5
Population Size	50	#bits of SLP	5
#bits for Weight Part	100	Crossover Rate	0.8
Investment Capital	100,000	Mutation Rate	0.03
Bounds of TPP	15%	Inversion Rate	0.6
Bounds of SLP	− 15%	#generation	100

6.1 Dataset descriptions

The three datasets with different stock trends used for experimental evaluations including uptrend, sideway trend and downtrend are described in this section. The time period from 2011/01 to 2016/12 of the stock prices were collected. The stock price series of them are illustrated in Figs. 8, 9 and 10.

In Fig. 8, we can observe that the stock prices are between 50 and 200. When using the buy and hold trading strategy (BHS), the returns are calculated as 29%, 8%, 32%, 1% and 27% for 2012, 2013, 2014, 2015 and 2016. The stock price of the sideway trend dataset in Fig. 9 is between 100 and 400. Again, when using BHS, the returns of the years from 2012 to 2016 are 32%, 25%, 10% and -21%. The downtrend stock price series showed in Fig. 10, the highest and lowest values are around 1500 and 100. The returns can be calculated using the BHS as −40%, −53%, −1%, −45% and −2% for 2012, 2013, 2014, 2015 and 2016. To generate candidate TSs, the used ten technical indicators are selected based on the literatures [5]14, and are shown in Table 11.

In accordance with the parameter setting used in [22], the generated trading rules with appropriate parameters for finding trading signals, including selling and buying signals, are given in Table 12.

Table 12 lists twenty rules, and ten of them are selling rules and others are buying rules. Then, a TS can be formed by selecting a buying rule and a selling rule. Take S_1 and B_1 as an example. The generated TS is "Buying signal: MA5 cross over MA20; Selling signal: MA5 cross down MA20". In the same way, totally 100 TSs can be generated. Based on the candidate TSs generation procedure, *top-m* TSs will be then selected for obtaining a GTSP and its SLTP according to the used ranking function.

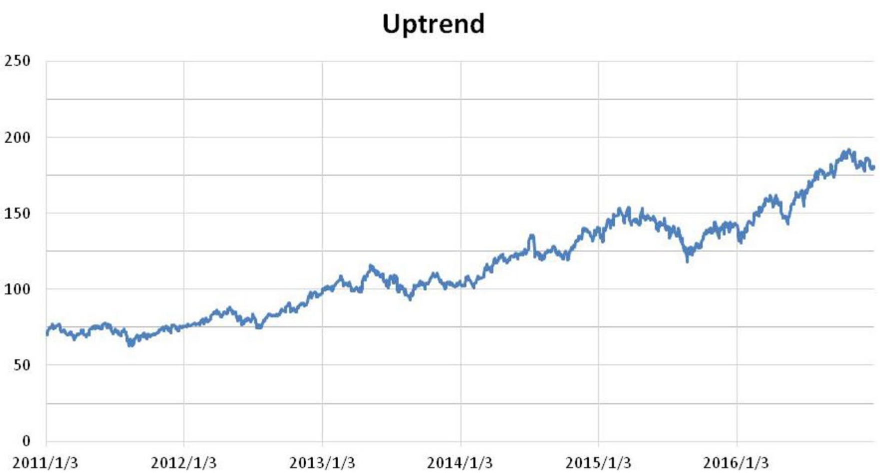

Fig. 8 Uptrend stock price series

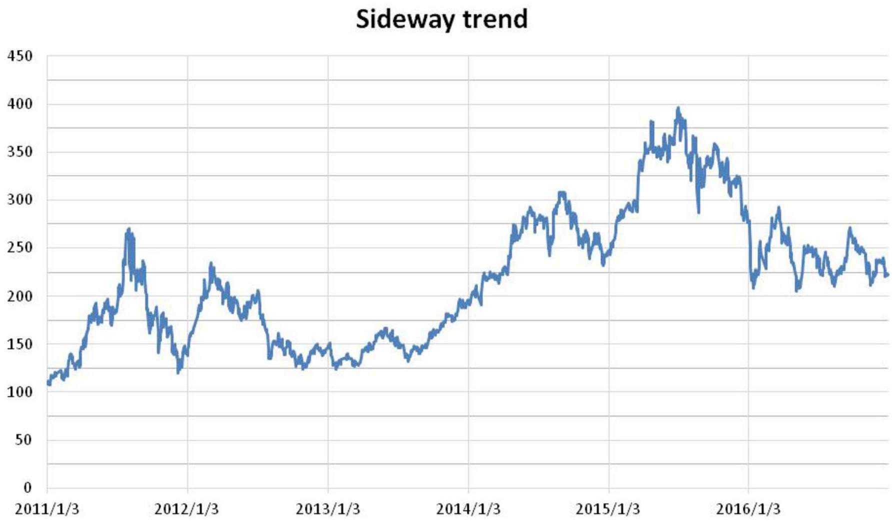

Fig. 9 Sideway trend stock price series

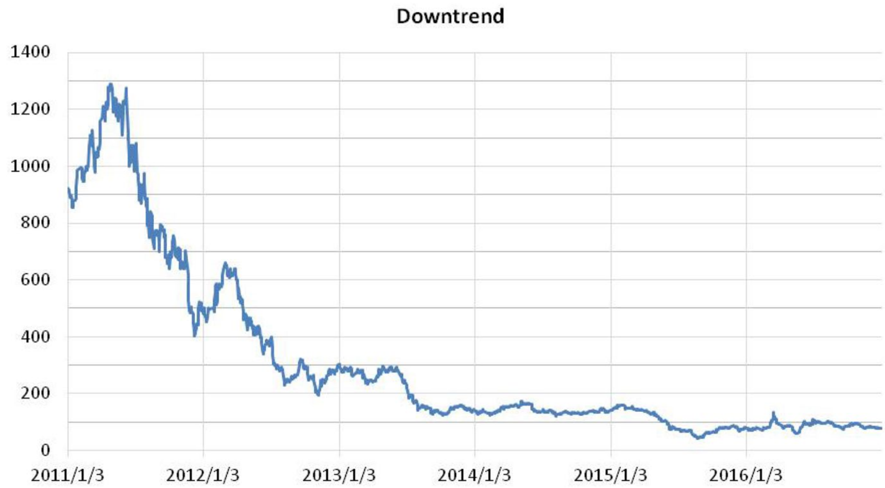

Fig. 10 Downtrend stock price series

6.2 Experimental evaluations on the datasets with different trends

Experiments were made on the three kinds of trends to show the effectiveness of the GTSP-SLTP algorithm. The three datasets are stock prices series with uptrend, sideway trend, and downtrend. The results for the three datasets are shown and discussed in Sects. 6.2.1, 6.2.2 and 6.2.3. To form needed TSs for generating initial population, two different ranking function selection strategies

Table 11 The used ten technical indicators

Id	Indicator	Id	Indicator
1	Moving Average (MA)	6	Commodity Channel Index (CCI)
2	Relative Strength Index (RSI)	7	Stochastic oscillator (KD)
3	Williams%R (WMS%R)	8	Moving Average Convergence-Divergence (MACD)
4	Momentum (MOM)	9	Bias ratio (BIAS)
5	Psychology (PSY)	10	Directional Movement Index (DMI)

Table 12 The trading rules with appropriate parameters

B#	Buying rules	*S#*	Selling rule
B_1	MA5 ↗ MA20	S_1	MA5 ↘ MA20
B_2	RSI ↗ 30	S_2	RSI ↘ 70
B_3	WMS%R ↘ 80	S_3	WMS%R ↗ 20
B_4	MOM ↗ 0	S_4	MOM ↘ 0
B_5	PSY ↗ 25%	S_5	PSY ↘ 75%
B_6	CCI ↗ -100	S_6	CCI ↘ 100
B_7	D < 20, K ↗ D	S_7	D > 80,K ↘ D
B_8	DIF ↗ MACD(DEM), or DIF ↗ 0	S_8	DIF ↘ MACD(DEM), or DIF ↘ 0
B_9	BIAS ↗ -4.5%	S_9	BIAS ↘ 5%
B_{10}	+DI ↗ -DI	S_{10}	+DI ↘ -DI

are adopted in the following experiments. The first selection strategy uses only average return rate as ranking function to select top-15 TSs out of 100 candidate TSs, and named TOP15R. The second selection strategy uses the three ranking functions to select the 15 TSs, and named TOP5R5F5D. In other words, it used the average return rate as the first ranking function to select top-5 TSs, the trading frequent as the second ranking function to select another top-5 TSs, and maximum draw down as the third ranking function to select last top-5 TSs. Then, the TSs generated using the TOP15R and TOP5R5F5D are utilized to find the GTSPs and its SLTPs based on the given training and testing datasets.

6.2.1 Evaluations on the uptrend dataset

In the following, experiments were made on the uptrend dataset to verify the effectiveness of the GTSP-SLTP algorithm on different training and testing periods. Comparison results of the GTSP-SLTP and BHS are given in Sect. 6.2.1.1. Then, comparison results of the GTSP-SLTP and the previous approach with predefined SLTPs [4] are stated in Sect. 6.2.1.2.

6.2.1.1 Comparison results of the proposed approach and BHS on the uptrend dataset Table 13 shows the comparison results of the GTSP-SLTP algorithm with the two ranking function selection strategies, TOP15R and TOP5R5F5D, and BHS

Table 13 Comparison results of the GTSP-SLTP and BHS on different training and testing periods for the uptrend dataset

Training Period	Testing Period		TOP15R	TOP5R5F5D	BHS
2011	2012	AgR	0.10	0.08	
		MaR	0.13	0.11	**0.28**
		MiR	0.06	0.05	
2012	2013	AgR	0.03	−0.03	
		MaR	**0.11**	0.05	0.07
		MiR	−0.04	−0.07	
2013	2014	AgR	0.14	0.13	
		MaR	0.22	0.16	**0.32**
		MiR	0.09	0.11	
2014	2015	AgR	−0.06	−0.05	
		MaR	−0.03	**0.00**	**0.00**
		MiR	−0.11	−0.11	
2015	2016	AgR	0.11	0.13	
		MaR	0.17	0.18	**0.26**
		MiR	0.05	0.08	
2011–2012	2013	AgR	0.05	0.02	
		MaR	**0.14**	0.05	0.07
		MiR	0.00	−0.05	
2012–2013	2014	AgR	0.17	0.21	
		MaR	0.24	0.27	**0.32**
		MiR	0.03	0.00	
2013–2014	2015	AgR	−0.06	−0.07	
		MaR	−0.02	−0.01	**0.00**
		MiR	−0.12	−0.13	
2014–2015	2016	AgR	*0.19*	0.18	
		MaR	*0.23*	0.23	**0.26**
		MiR	*0.16*	0.16	
2011–2013	2014	AgR	0.09	0.12	
		MaR	0.12	0.21	**0.32**
		MiR	0.07	0.07	
2012–2014	2015	AgR	−0.10	−0.10	
		MaR	−0.05	−0.01	**0.00**
		MiR	−0.16	−0.16	
2013–2015	2016	AgR	0.15	0.19	
		MaR	0.20	0.22	**0.26**
		MiR	0.11	0.16	

Bold values indicate the hightest returns for the given training periods

on different training and testing periods in terms of average, maximum, and minimum returns that are abbreviated to AgR, MaR and MiR.

Table 13 shows the BHS is basically better than the GTSP-SLTP algorithm in terms of returns. But we observed that when the training period in 2012 and 2011–2012, and the testing period in 2013, the MaR values of the optimized GTSP are 0.11 and 0.14; they are better than BHS. In addition, we can also see that when the training period is two years, the GTSP-SLTP can reach the highest returns than other training periods. For example, the AgR values of the testing period 2016 of the GTSP-SLTP algorithm with TOP15R in the two-years training periods is 19% which is better than 11% in one-year and 15% for the three-years training periods. It can also be observed that when using three-year training periods, the ranking function selection strategy TOP5R5F5D is better than TOP15R in terms of returns. For instance, the AgR, MaR and MiR values of the testing period 2016 for the TOP5R5F5D are 19%, 22% and 16%; they are better than TOP15R that are 15%, 20% and 11%. The results indicated that when users prefer long-term investment, the ranking function selection strategies TOP5R5F5D is suggested for generating candidate trading strategies. Overall speaking, although returns of the GTSP-SLTP algorithm are negative in few testing periods, it still can obtain positive returns in the most testing periods.

6.2.1.2 Comparison results of proposed and previous approaches on the uptrend dataset Table 14 shows the comparison results of the GTSP-SLTP algorithm and the previous approach with predefined SLTPs [4] on different training and testing periods in terms of AgR, MaR and MiR. For the previous approach, the SLTPs in the given range that can reach the largest returns in training periods were set and the optimized GTSPs are used to compare with that generated by the proposed approach. The two values show in the parentheses are the stop-loss point and take-profit point of the previous and the proposed approaches, and the two ranking function selection strategies, the TOP15R and TOP5R5F5D, are used in the two approaches.

Table 14 shows that excepting the TOP5R5F5D on the testing period 2013, the returns of the GTSP-SLTP algorithm obtains better results than the previous approach. In addition, we can also observe that the MiR value of the GTSP-SLTP algorithm is basically better than the previous approach. For instance, the MiR values of the previous approach in testing period 2015 are both negative 20% for the TOP15R and TOP5R5F5D. However, the MiR values of the GTSP-SLTP algorithm are both negative 11% for the two cases. In other words, the results reveal that the GTSP-SLTP algorithm has a better ability to reduce loss than the previous approach.

6.2.2 Evaluations on the sideway trend dataset

In this section, experiments were made on the second dataset to verify the performance of the GTSP-SLTP algorithm. In the following, comparison results of

Table 14 Comparison results of the GTSP-SLTP and the previous approach with predefined SLTPs on different training and testing periods for the uptrend dataset

Training period	Testing period		Previous approach with predefined SLTP			GTSP− SLTP		
2011	2012	TOP15R	(− 15% ~ 5%)			(− 14% ~ 6%)		
			AgR	MaR	MiR	AgR	MaR	MiR
			0.07	0.10	0.05	**0.10**	0.13	0.06
		TOP5R5F5D	(− 15% ~ 5%)	(− 15% ~ 6%)				
			AgR	MaR	MiR	AgR	MaR	MiR
			0.07	0.11	0.05	**0.08**	0.11	0.05
2012	2013	TOP15R	(− 5% ~ 15%)	(− 15% ~ 11%)				
			AgR	MaR	MiR	AgR	MaR	MiR
			− 0.04	0.03	− 0.07	**0.03**	0.11	− 0.04
		TOP5R5F5D	(− 5% ~ 15%)	(− 2% ~ 13%)				
			AgR	MaR	MiR	AgR	MaR	MiR
			− 0.02	0.15	− 0.07	− 0.03	0.05	− 0.07
2013	2014	TOP15R	(− 10% ~ 10%)	(− 9% ~ 13%)				
			AgR	MaR	MiR	AgR	MaR	MiR
			0.12	0.13	0.11	**0.14**	0.22	0.09
		TOP5R5F5D	(− 10% ~ 10%)	(− 10% ~ 13%)				
			AgR	MaR	MiR	AgR	MaR	MiR
			0.12	0.13	0.11	**0.13**	0.16	0.11
2014	2015	TOP15R	(− 15% ~ 15%)	(− 1% ~ 15%)				
			AgR	MaR	MiR	AgR	MaR	MiR
			− 0.13	− 0.02	− 0.20	**− 0.06**	− 0.03	− 0.11
		TOP5R5F5D	(− 10% ~ 15%)	(− 1% ~ 15%)				
			AgR	MaR	MiR	AgR	MaR	MiR
			− 0.14	− 0.01	− 0.20	**− 0.05**	0.00	− 0.11
2015	2016	TOP15R	(− 15% ~ 5%)	(− 3% ~ 11%)				
			AgR	MaR	MiR	AgR	MaR	MiR
			0.07	0.08	0.05	**0.11**	0.17	0.05
		TOP5R5F5D	(− 15% ~ 5%)	(0% ~ 15%)				
			AgR	MaR	MiR	AgR	MaR	MiR
			0.07	0.09	0.06	**0.08**	0.08	0.08

Bold values indicate the hightest returns for the given training periods

the GTSP-SLTP algorithm and BHS are given in Sect. 6.2.2.1. Then, comparison results of the GTSP-SLTP algorithm and the previous approach with predefined SLTPs [4] are stated in Sect. 6.2.2.2.

6.2.2.1 Comparison results of the proposed approach and BHS on the sideway trend dataset The experiments were conducted to show the comparison results of the GTSP-SLTP algorithm with TOP15R and TOP5R5F5D and BHS on different training and testing periods in terms of AgR, MaR and MiR. The results are shown in Table 15.

Table 15 Comparison returns of the proposed approach and BHS on the sideway trend dataset

Training period	Testing period		TOP15R	TOP5R5F5D	BHS
2011	2012	AgR	−0.10	−0.11	
		MaR	0.00	−0.01	**0.02**
		MiR	−0.16	−0.16	
2012	2013	AgR	0.07	0.06	
		MaR	0.12	0.10	**0.34**
		MiR	0.04	0.03	
2013	2014	AgR	0.16	0.21	
		MaR	**0.34**	**0.39**	0.26
		MiR	0.01	0.06	
2014	2015	AgR	0.25	0.20	
		MaR	**0.45**	**0.44**	0.11
		MiR	0.00	−0.03	
2015	2016	AgR	0.10	0.09	
		MaR	**0.16**	**0.16**	−0.20
		MiR	0.02	0.02	
2011–2012	2013	AgR	0.15	0.14	
		MaR	0.20	0.20	**0.34**
		MiR	0.06	0.05	
2012–2013	2014	AgR	0.17	0.08	
		MaR	**0.31**	0.12	0.26
		MiR	0.05	0.04	
2013–2014	2015	AgR	0.24	0.18	
		MaR	**0.45**	**0.45**	0.11
		MiR	−0.06	−0.12	
2014–2015	2016	AgR	0.12	0.14	
		MaR	**0.21**	**0.22**	−0.20
		MiR	0.00	−0.04	
2011–2013	2014	AgR	0.26	0.23	
		MaR	**0.43**	**0.39**	0.26
		MiR	0.04	0.04	
2012–2014	2015	AgR	0.14	0.18	
		MaR	**0.40**	**0.45**	0.11
		MiR	−0.11	−0.07	
2013–2015	2016	AgR	0.15	0.18	
		MaR	**0.21**	**0.25**	−0.20
		MiR	0.02	0.02	

Bold values indicate the hightest returns for the given training periods

From the Table 15, we can observe that it shows that the BHS is better than the GTSP-SLTP only in few testing periods, e.g., the testing periods are 2012 and 2013 when the training period is one year, the testing period is 2013 when the training period is 2011 to 2012. For other testing periods, the GTSP-SLTP algorithm can find higher returns than BHS. Especially in testing period 2016, the GTSP-SLTP algorithm can reach at least 10% returns in different training periods while the return of BHS is -20%. These results indicate that the GTSP-SLTP algorithm is effective when the trend of a stock price series is sideway trend.

6.2.2.2 Comparison results of the proposed and previous approaches on the sideway trend dataset Experiments were then made to show the effectiveness of the GTSP-SLTP algorithm via comparing to the previous approach with predefined SLTPs on the sideway dataset. The experimental results are shown in Table 16.

From Table 16, we have two observations. The first one is that the AgR values of the GTSP-SLTP algorithm are higher than the previous approach. Take testing period 2016 as an example, we can see that the AgR values of the GTSP-SLTP with TOP15R and TOP5R5F5D are 10% and 9% that are higher than 5% and 8% compared to the previous approach. The second observation is that the GTSP-SLTP is also better than the previous approach in terms of the ability to reduce loss. Take testing period 2012 as an example, the MiR values of the GTSP-SLTP are both -16% for TOP15R and TOP5R5F5D that are smaller than -36% and -30% of the previous approach. Based on the observations, they indicate that the GTSP-SLTP algorithm is effective on the sideway dataset.

6.2.3 Evaluations on the downtrend dataset

In this section, experiments were conducted on the stock prices series with downtrend to show the merits of the GTSP-SLTP algorithm. In the following, comparison results of the GTSP-SLTP algorithm and BHS, and the previous approach with predefined SLTPs [4], are shown in Sects. 6.2.3.1 and 6.2.3.2.

6.2.3.1 Comparison results of the proposed approach and BHS on the downtrend dataset Experiments on the downtrend dataset to verify the effectiveness of the GTSP-SLTP algorithm is critical because to avoid massive loss is always the important purpose in the market, especially in bear markets. The comparison results of the GTSP-SLTP algorithm with TOP15R and TOP5R5F5D and BHS on different training and testing periods in terms of AgR, MaR and MiR are shown in Table 17.

From Table 17, it first shows that the returns of BHS are negative in all testing periods on the downtrend dataset. In other words, the BHS is not useful on the downtrend dataset. On the other hand, we can observe the AgR, MaR, and MiR values of the GTSP-SLTP algorithm are significantly better than BHS. Take testing period 2015 and the training period is 2012 to 2014 as an example. the return of the BHS is −45%. However, the AgR values of the GTSP-SLTP are both -13% for the TOP15R and TOP5R5F5D. In addition, we also observe that the GTSP-SLTP algorithm with TOP15R is better than it with TOP5R5F5D. Take testing period 2015 over different training periods as an example, the AgR values of the

Table 16 Comparison returns of the GTSP-SLTP algorithm and the previous approach with SLTPs on the sideway trend dataset

Training Period	Testing Period		Previous approach with predefined SLTP			GTSP− SLTP		
2011	2012	TOP15R	(− 10% ~ 15%)			(− 3% ~ 15%)		
			AgR	MaR	MiR	AgR	MaR	MiR
			− 0.25	0.00	− 0.36	**− 0.10**	0.00	− 0.16
		TOP5R5F5D	(− 10% ~ 15%)	(0% ~ 8%)				
			AgR	MaR	MiR	AgR	MaR	MiR
			− 0.18	− 0.01	− 0.30	**− 0.11**	− 0.01	− 0.16
2012	2013	TOP15R	(− 0% ~ 0%)	(− 5% ~ 1%)				
			AgR	MaR	MiR	AgR	MaR	MiR
			0.04	0.07	0.01	**0.07**	0.12	0.04
		TOP5R5F5D	(− 0% ~ 0%)	(0% ~ 3%)				
			AgR	MaR	MiR	AgR	MaR	MiR
			0.04	0.07	0.01	**0.06**	0.10	0.03
2013	2014	TOP15R	(− 15% ~ 15%)	(− 1% ~ 11%)				
			AgR	MaR	MiR	AgR	MaR	MiR
			0.24	0.38	0.13	0.16	0.34	0.01
		TOP5R5F5D	(− 10% ~ 5%)	(− 1% ~ 15%)				
			AgR	MaR	MiR	AgR	MaR	MiR
			0.14	0.28	0.03	**0.21**	0.39	0.06
2014	2015	TOP15R	(− 15% ~ 15%)	(− 11% ~ 13%)				
			AgR	MaR	MiR	AgR	MaR	MiR
			0.27	0.44	0.06	0.25	0.45	− 0.00
		TOP5R5F5D	(− 10% ~ 15%)	(− 7% ~ 10%)				
			AgR	MaR	MiR	AgR	MaR	MiR
			0.27	0.45	0.00	0.20	0.44	− 0.03

Table 16 (continued)

Training Period	Testing Period		Previous approach with predefined SLTP			GTSP−SLTP		
2015	2016	TOP15R	(−0%~0%)	(−9%~13%)				
			AgR	MaR	MiR	AgR	MaR	MiR
			0.05	0.13	−0.04	**0.10**	0.16	0.02
		TOP5R5F5D	(−15%~5%)	(−15%~13%)				
			AgR	MaR	MiR	AgR	MaR	MiR
			0.08	0.16	−0.05	**0.09**	0.16	0.02

Bold values indicate the hightest returns for the given training periods

Table 17 Comparison returns of the proposed approach and BHS on the downtrend dataset

Training period	Testing Period		TOP15R	TOP5R5F5D	BHS
2011	2012	AgR	−0.24	−0.22	
		MaR	**−0.07**	**−0.08**	−0.41
		MiR	−0.51	−0.42	
2012	2013	AgR	−0.10	−0.07	
		MaR	**−0.01**	**−0.01**	−0.54
		MiR	−0.20	−0.12	
2013	2014	AgR	0.07	−0.02	
		MaR	**0.15**	**0.03**	−0.01
		MiR	−0.04	−0.07	
2014	2015	AgR	−0.10	−0.18	
		MaR	**0.07**	**0.07**	−0.45
		MiR	−0.52	−0.50	
2015	2016	AgR	0.04	0.02	
		MaR	**0.10**	**0.16**	−0.02
		MiR	−0.01	−0.06	
2011–2012	2013	AgR	−0.09	−0.10	
		MaR	**0.00**	**0.00**	−0.54
		MiR	−0.24	−0.20	
2012–2013	2014	AgR	−0.02	−0.02	
		MaR	**−0.01**	**−0.01**	**−0.01**
		MiR	−0.03	−0.03	
2013–2014	2015	AgR	−0.36	−0.51	
		MaR	**−0.10**	**−0.34**	−0.45
		MiR	−0.59	−0.60	
2014–2015	2016	AgR	0.03	0.04	
		MaR	**0.11**	**0.10**	−0.02
		MiR	−0.03	−0.01	
2011–2013	2014	AgR	−0.02	−0.02	
		MaR	**−0.01**	**−0.01**	**−0.01**
		MiR	−0.04	−0.03	
2012–2014	2015	AgR	−0.13	−0.13	
		MaR	**−0.06**	**−0.09**	−0.45
		MiR	−0.23	−0.18	
2013–2015	2016	AgR	0.04	0.05	
		MaR	**0.08**	**0.09**	−0.02
		MiR	0.00	0.01	

Bold values indicate the hightest returns for the given training periods

GTSP-SLTP with TOP15R on one-year, two-years and three-years training periods are −10%, −36 and −13% are higher than −18%, −51% and −13% by the GTSP-SLTP with TOP5R5F5D. The experimental results indicate that the GTSP-SLTP algorithm is effective in reducing massive loss.

Table 18 Comparison returns of the GTSP-SLTP algorithm and the previous approach with SLTPs on the downtrend dataset

Training period	Testing period		Previous approach with predefined SLTPs			GTSP−SLTP		
2011	2012	**TOP15R**	(−10% ~ 15%)			(−12% ~ 13%)		
			AgR	MaR	MiR	AgR	MaR	MiR
			−0.27	−0.07	−0.53	**−0.24**	−0.07	−0.51
		TOP5R5F5D	(−15% ~ 15%)	(−2% ~ 12%)				
			AgR	MaR	MiR	AgR	MaR	MiR
			−0.32	−0.11	−0.55	**−0.22**	−0.08	−0.42
2012	2013	**TOP15R**	(−0% ~ 0%)	(−1% ~ 9%)				
			AgR	MaR	MiR	AgR	MaR	MiR
			−0.06	−0.02	−0.12	−0.10	−0.01	−0.20
		TOP5R5F5D	(−5% ~ 10%)	(−1% ~ 12%)				
			AgR	MaR	MiR	AgR	MaR	MiR
			−0.08	−0.04	−0.16	**−0.07**	−0.01	−0.12
2013	2014	**TOP15R**	(−10% ~ 5%)	(−15% ~ 7%)				
			AgR	MaR	MiR	AgR	MaR	MiR
			0.05	0.11	−0.04	**0.07**	0.15	−0.04
		TOP5R5F5D	(−15% ~ 5%)	(−3% ~ 8%)				
			AgR	MaR	MiR	AgR	MaR	MiR
			0.05	0.15	−0.04	−0.02	0.03	−0.07
2014	2015	**TOP15R**	(−15% ~ 15%)	(−7% ~ 15%)				
			AgR	MaR	MiR	AgR	MaR	MiR
			−0.16	0.10	−0.55	**−0.10**	0.07	−0.52
		TOP5R5F5D	(−15% ~ 15%)	(−13% ~ 15%)				
			AgR	MaR	MiR	AgR	MaR	MiR
			−0.22	0.12	−0.57	**−0.18**	0.07	−0.50

Table 18 (continued)

Training period	Testing period		Previous approach with predefined SLTPs			GTSP−SLTP		
2015	2016	**TOP15R**	(−15% ~ 15%)	(−6% ~ 15%)				
			AgR	MaR	MiR	AgR	MaR	MiR
			0.08	0.22	−0.04	0.04	0.10	−0.01
		TOP5R5F5D	(−15% ~ 15%)	(−15% ~ 13%)				
			AgR	MaR	MiR	AgR	MaR	MiR
			0.03	0.18	−0.05	0.02	0.16	−0.06

Bold values indicate the hightest returns for the given training periods

6.2.3.2 Comparison results of the proposed and previous approaches on the downtrend dataset To comparing the GTSP-SLTP algorithm to the previous approach with the predefined SLTPs, the experiments on different training and testing periods were also made for the evaluation. The comparison results of the GTSP-SLTP algorithm and the previous approach with the TOP15R and TOP5R5F5D in terms of AgR, MaR and MiR are shown in Table 18.

From Table 18, the six out of ten AgR values of the GTSP-SLTP algorithm are higher than the previous approach. Take the testing period 2015 as an example. The AgR values of the GTSP-SLTP algorithm are −10% and −18% for the TOP15R and TOP5R5F5D that are better than −16% and −22% of the previous approach. Comparing the MiR values of the GTSP-SLTP algorithm and the previous approach, we can also know that the GTSP-SLTP algorithm can reach similar or even a little better MiR values than the previous approach. For instance, take the testing period 2016 as an example. The MiR value of the GTSP-SLTP are −1% for the TOP15R which is better than -4% of the previous approach. In other words, the results reveal that the GTSP-SLTP algorithm can automatically find the GTSP and suitable SLTPs for reducing loss.

6.3 Case study on a group stock portfolio

To show the merits of the GTSP-SLTP algorithm, we applied it on the group stock portfolio (GSP) which is a type of stock portfolio that can be generated by the algorithm presented in [6]. Since a GSP is composed of stock groups, many stock portfolios can be generated. This case study is conducted on a real financial dataset with 31 companies that are collected from 2010/01 to 2012/12. We first used 2 years dataset 2010–2011 as the training period to generate ten GSPs, and used one-year dataset 2012 as the testing period for the comparison. The GTSP-SLTP algorithm with TOP15R and

Table 19 Comparison results of the generated GSPs with BHS and with the GTSP-SLTP algorithm

Id	MiR of GSP with BHS	MiR of GSP with GTSP− SLTP	
		TOP15R	TOP5R5F5D
1	0.521	0.222	0.220
2	0.551	0.240	0.237
3	**−0.006**	**0.011**	**0.006**
4	**−0.009**	**0.010**	**0.005**
5	0.554	0.242	0.241
6	**−0.003**	**0.013**	**0.010**
7	0.541	0.234	0.223
8	**−0.005**	**0.003**	**0.000**
9	0.561	0.245	0.233
10	**−0.007**	**0.008**	**0.007**
Avg. MiR	**0.2698**	0.1228	0.1182

Bold values indicate the hightest returns for the given training periods

TOP5R5F5D has then been employed to obtain suitable GTSP and SLTP for each company. The comparison results of the generated GSPs with BHS and the GTSP-SLTP algorithm with TOP15R and TOP5R5F5D in terms of MiR are shown in Table 19.

Table 19 shows that the average MiR value of the GSP with BHS and with GTSP-SLTP with TOP15R and TOP5R5F5D are 26.98%, 12.28% and 11.82% that indicate the GSP with BHS is better than the GSP with the GTSP-SLTP. However, we also observe that the MiR values of the GSP with the GTSP-SLTP are always positive which means that using the GTSP-SLTP algorithm has a higher ability to avoid risk than using the BHS. To verify whether the GSP with the GTSP-SLTP algorithm can reach more stable returns clearly, comparison results of the GSP with BHS and with GTSP-SLTP in terms of variance of returns are shown in Table 20.

From Table 20, we can easily see that the average values of variance of returns of GSP with BHS and with the GTSP-SLTP using TOP15R and TOP5R5F5D are 3.63%, 0.061% and 0.57%, respectively. These results reveal that the GSP with the GTSP-SLTP can actually provide more robust returns than that with BHS. In other words, by using the proposed algorithm, the advantage is that the appropriate trading signals generated by various trading strategies can be suggested for reaching a better trading performance. Through this case study, we can conclude that the GTSP-SLTP algorithm can provide a more safety way for users making investment plans.

7 Conclusions and future work

Trading strategy is commonly used to find trading signals in the markets. Since different trading strategies have their functionalities, investors prefer to have a trading strategy portfolio instead of a trading strategy for making the more

Table 20 Variance of returns of the GSP with BHS and with GTSP-SLTP

Id	Variance of Return of GSP with BHS	Variance of Return of GSP with GTSP-SLTP	
		TOP15R	**TOP5R5F5D**
1	0.00023	0.00009	0.00010
2	0.00009	0.00003	0.00005
3	0.08011	0.01339	0.01234
4	0.08042	0.01340	0.01238
5	0.00011	0.00003	0.00003
6	0.07427	0.01238	0.01146
7	0.00018	0.00006	0.00006
8	0.07395	0.01230	0.01133
9	0.00006	0.00002	0.00002
10	0.05416	0.00943	0.00949
Average Value	0.03635	**0.00611**	**0.00572**

Bold values indicate the hightest returns for the given training periods

profitable investment plans along with appropriate stop-loss and take-profit points. To provide a reliable mechanism for suggesting various trading strategy portfolios and stop-loss and take-profit points, the GTSP-SLTP algorithm has been proposed to reach the goal. Empirical experiments on three datasets showed that: (1) When comparing to the BHS, the results show that the GTSP-SLTP algorithm is effective on sideway trend and downtrend datasets. In other words, the GTSP-SLTP algorithm is effective in reducing massive loss; (2) Comparing to the previous approach, the results also show that the GTSP-SLTP algorithm can reach a higher return than the previous approach; (3) Furthermore, the case study reveals that when the GTSP-SLTP algorithm is employed to obtain trading signals of a given group stock portfolio (GSP), the variance of returns of the GSP with the trading signals are smaller than that without the trading signals. Two numerical values are listed as follows: (1) To avoid massive loss in bear markets, experiments on the downtrend dataset showed that using 2015 as testing and 2012 to 2014 as training periods, the return of the BHS is -45%. However, by using the GTSP-SLTP algorithm, the loss is reduced to – 13%; (2) To generate appropriate trading signals for assets, the experimental results indicate that average values of variance of returns of portfolio with the GTSP-SLTP are between 0.061% and 0.57% by the designed GTSP-SLTP algorithm for portfolio management. In the future, we will continue to enhance the proposed approach in following directions, e.g., considering other indicators to construct more candidate trading strategies and utilizing multi-objective genetic algorithms to obtain more diverse solutions.

Acknowledgment This research was supported by the Ministry of Science and Technology of the Republic of China under grants MOST 106-2221-E-032 -049 -MY2 and MOST 108-2221-E-027-119.

Funding Open access funding provided by Western Norway University Of Applied Sciences.

References

1. Chang, Y., & Lee, M. (2016). Incorporating Markov decision process on genetic algorithms to formulate trading strategies for stock markets. *Applied Soft Computing, 52*(10), 1143–1153.
2. Chang, T. J., Yang, S. C., & Chang, K. J. (2009). Portfolio optimization problems in different risk measures using genetic algorithm. *Expert Systems with Applications, 36,* 10529–10537.
3. Chang-Chien, Y. W., & Chen, Y. L. (2010). Mining associative classification rules with stock trading data–A GA-based method. *Knowledge-Based Systems, 23*(6), 605–614.
4. Chen, C. H., Chen, Y. H., Lin, J. C. W., & Wu, M. E. (2019). An effective approach for obtaining a group trading strategy portfolio using grouping genetic algorithm. *IEEE Access, 7,* 7313–7325.

5. Chen, J., Hou, J., Wu, S., & Chang-Chien, Y. (2009). Constructing investment strategy portfolios by combination genetic algorithms. *Expert Systems with Applications, 36*(2), 3824–3828.
6. Chen, C. H., Lu, C. Y., & Lin, C. B. (2020). An intelligence approach for group stock portfolio optimization with a trading mechanism. *Knowledge and Information Systems, 62,* 287–316.
7. Chen, C. H., Lu, C. Y., Hong, T. P., Lin, J. C. W., & Gaeta, M. (2019). An effective approach for the diverse group stock portfolio optimization using grouping genetic algorithm. *IEEE Access, 7,* 155871–155884.
8. C. H. Chen, W. Y. Shen, T. P. Hong and J. H. Su, "An island-based algorithm for group stock portfolio optimization," *Joint 17th World Congress of International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems*, pp. 1–5, 2017
9. Chou, Y. H., Kuo, S. Y., Chen, C. Y., & Chao, H. C. (2014). A rule-based dynamic decision-making stock trading system based on quantum-inspired tabu search algorithm. *IEEE Access, 2,* 883–896.
10. Chou, Y. H., Kuo, S. Y., & Lo, Y. T. (2017). Portfolio optimization based on funds standardization and genetic algorithm. *IEEE Access, 5,* 21885–21900.
11. Drezewski, R., Dziuban, G., & Pajak, K. (2018). The bio-inspired optimization of trading strategies and its impact on the efficient market hypothesis and sustainable development strategies. *Sustainability, 10*(51460), 1–45.
12. Falkenauer, E. (1994). A new representation and operators for genetic algorithms applied to grouping problems. *Evolutionary Computation, 2,* 123–144.
13. Falkenauer, E. (1996). A hybrid grouping genetic algorithm for bin packing. *Journal of Heuristics, 2,* 5–30.
14. Fu, T. C., Chung, C. P., & Chung, F. L. (2013). Adopting genetic algorithms for technical analysis and portfolio management. *Computers and Mathematics with Applications, 66*(10), 1743–1757.
15. Faias, J. A., & Santa-Clara, P. (2017). Optimal option portfolio strategies: Deepening the puzzle of index option mispricing. *Journal of Financial and Quantitative Analysis, 52*(1), 277–303.
16. Goldberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning. *Addison Wesley, 1989,* 36.
17. Grefenstette, J. J. (1986). Optimization of control parameters for genetic algorithms. *IEEE Transactions on System Man, and Cybernetics, 16,* 122–128.
18. L. R. Z. Hoklie, "Resolving multi objective stock portfolio optimization problem using genetic algorithm," *International Conference on Computer and Automation Engineering*, pp. 40–44, 2010.
19. Holland, J. H. (1975). *Adaptation in natural and artificial systems*. University of Michigan Press.
20. Ha, Y., & Zhang, H. (2020). Algorithmic trading for online portfolio selection under limited market liquidity. *European Journal of Operational Research, 286*(3), 1033–1051.
21. Kaminski, K. M., & Lo, A. W. (2014). When do stop-loss rules stop losses? *Journal of Financial Markets, 18,* 234–254.
22. Kim, Y., & Enke, D. (2016). Developing a rule change trading system for the futures market using rough set analysis. *Expert Systems with Applications, 59,* 165–173.
23. Kumar, R., & Bhattacharya, S. (2012). Cooperative search using agents for cardinality constrained portfolio selection problem. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 42,* 1510–1518.
24. S. Y. Kuo, C. Kuo and Y. H. Chou, "Dynamic stock trading system based on quantum-inspired tabu search algorithm," *The IEEE Congress on Evolutionary Computation*, pp. 1029–1036, 2013.
25. Y. Leu and T. I. Chiu, "An effective stock portfolio trading strategy using genetic algorithms and weighted fuzzy time series," *The North-East Asia Symposium on Nano, Information Technology and Reliability*, pp. 70–75, 2011.
26. Liu, Y. J., & Zhang, W. G. (2013). Fuzzy portfolio optimization model under real constraints. *Insurance: Mathematics and Economics, 53*(3), 704–711.
27. Lo, A. W., & Remorov, A. (2017). Stop-loss strategies with serial correlation, regime switching, and transaction costs. *Journal of Financial Markets, 34,* 1–15.
28. Lwin, K., Qu, R., & Kendall, G. (2014). A learning-guided multi-objective evolutionary algorithm for constrained portfolio optimization. *Applied Soft Computing, 24,* 757–772.
29. Lin, T. Y., Chen, C. W. S., & Syu, F. Y. (2020). Multi-asset pair-trading strategy: A statistical learning approach. *The North American Journal of Economics and Finance, 55,* 101295.
30. Markowitz, H. (1952). Portfolio selection. *Journal of Finance, 7*(1), 77–91.
31. Markowitz, H. M. (2009). *Harry Markowitz: Selected works*. World Scientific Publishing Company.
32. Nuij, W., Milea, V., Hogenboom, F., Frasincar, F., & Kaymak, U. (2014). An automated framework for incorporating news into stock trading strategies. *IEEE Transactions on Knowledge and Data Engineering, 26*(4), 823–835.

33. Park, H., Sim, M. K., & Choi, D. G. (2020). An intelligent financial portfolio trading strategy using deep Q-learning. *Expert Systems with Applications, 158,* 113573.
34. X. Qin, H. Wang, F. Li, J. Chen, X. Zhou, X. Du, S. Wang, "Optimizing parameters of algorithm trading strategies using MapReduce," *International Conference on Fuzzy Systems and Knowledge Discovery*, pp. 2738–2741, 2012.
35. Wu, M. E., Wang, C. H., & Chung, W. H. (2017). Using trading mechanisms to investigate large futures data and their implications to market trends. *Soft Computing, 21*(11), 2821–2834.
36. Wu, X., Chen, H., Wang, J., Troiano, L., Loia, V., & Fujita, H. (2020). Adaptive stock trading strategies with deep reinforcement learning methods. *Information Sciences, 538,* 142–158.
37. Yao, H., Li, Z., & Li, D. (2016). Multi-period mean-variance portfolio selection with stochastic interest rate and uncontrollable liability. *European Journal of Operational Research, 252*(3), 837–851.

Authors and Affiliations

Chun-Hao Chen[1] · Yu-Hsuan Chen[2] · Vicente Garcia Diaz[3] · Jerry Chun-Wei Lin[4]

Chun-Hao Chen
chchen@ntut.edu.tw

Vicente Garcia Diaz
garciavicente@uniovi.es

[1] Department of Information and Finance Management, National Taipei University of Technology, Taipei, Taiwan

[2] Department of Computer Science and Information Engineering, Tamkang University, Taipei, Taiwan

[3] School of Computer Engineering, University of Oviedo, Oviedo, Spain

[4] Department of Computer Science, Electrical Engineering and Mathematical Sciences, Western Norway University of Applied Sciences, Bergen, Norway