IEEJ TRANSACTIONS ON ELECTRICAL AND ELECTRONIC ENGINEERING
IEEJ Trans 2011; **6**: 403–413
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI:10.1002/tee.20676

Paper

A Portfolio Selection Model Using Genetic Relation Algorithm and Genetic Network Programming

Yan Chen*, Non-member
Kotaro Hirasawa**[a], Member

In this paper, new evolutionary computation methods named genetic relation algorithm (GRA) and genetic network programming (GNP) have been applied to the portfolio selection problem. The number of brands in the stock market is generally very large, therefore, techniques for selecting the effective portfolio are likely to be of interest in the financial field. In order to pick up the most efficient portfolio, the proposed model considers the correlation coefficient between stock brands as strength, which indicates the relation between nodes in GRA. The algorithm evaluates the relationships between stock brands using a specific measure of strength and generates the optimal portfolio in the final generation. Then, the selected portfolio is further optimized by the stock trading model of GNP. In a sense, the proposed model is an integrated intelligent model. A comprehensive analysis of the results is provided, and it is clarified that the proposed model can obtain much higher profits than other traditional methods.

Keywords: portfolio selection, genetic relation algorithm, genetic network programming

Received 23 April 2009; Revised 2 January 2010

1. Introduction

This paper presents an application of the evolutionary computation methods named genetic relation algorithm (GRA) and genetic network programming (GNP) to the problem of portfolio selection in the financial field. In the conventional portfolio optimization problem, given the investor's objectives and economic conditions, we want to find out what assets to include in an optimal portfolio, in order to maximize the expected return and minimize risk simultaneously.

In order to solve this optimization problem, Markowitz [1–4] first proposed a mean-variance optimization model to design an optimum portfolio as the foundation of portfolio selection, which assumes that the total portfolio can be obtained using the mean return of the assets and the variance of the return over these assets. In the case of linear constraints, Best and Kale [5] and Stein *et al*. [6] solved the problem efficiently by parametric quadratic programming. However, there are many real-world nonlinear constraints which limit the number of different assets in a portfolio. As the number of brands in the stock market is generally very large, techniques for selecting the effective portfolio are likely to be of interest in the financial field.

In recent decades, various approaches in the artificial intelligence (AI) field have been applied to several financial problems, especially for the stock market activities. Generally speaking, with the increasing need for more efficient portfolio selection and stock trading models, AI approaches have been confirmed to outperform the conventional statistical models [7–10] in terms of that they overcome the limitation of assumptions [11,12].

As one of the most popular heuristic optimization techniques, genetic algorithms (GA) were originally developed by Holland [13]. Subsequently, GA had been applied to many optimization problems in engineering and the operations research [14–17]. Lin *et al*. [18] considered the multi-objective GA for the portfolio selection problem. Oh *et al*. [19] proposed a new portfolio selection algorithm based on portfolio beta by using GA. Genetic programming (GP), as an extension of GA, has been described by Koza [20]. So far, GP has been applied to a wide range of financial fields such as generating trading rules on the stock markets by Potvin *et al*. [21], Dempster and Jones [22] and Marney *et al*. [23], and bankruptcy prediction by Etemadi *et al*. [24], etc. However, when GA is applied to the portfolio optimization, the problem is that many chromosomes are coded into the same portfolio, or similar chromosomes may be coded into very different portfolios which makes it more difficult for GA to produce better chromosomes from good ones. These problems multiply the GA's search space and make GA less efficient in finding the optimal portfolio. Also, when GP is applied to the financial forecasting and portfolio selection problem, it occasionally causes some bloating problems for its tree structure. Similarly, artificial neural network (ANN) has also been widely used in the financial field [25]. Lam [26] applies the back-propagation algorithm to integrate fundamental and technical analysis for financial performance prediction. Fernandez and Gomez [11] present a different heuristic method based on ANN. Likewise, Yu *et al*. [27] introduce a new AI technique and propose a fast and efficient radial basis function (RBF) neural network-based methodology. However, as ANN is data-driven model, the underlying rules in the data are not always apparent, which leads to so-called black-box model, and investors cannot benefit from the knowledge discovery in the analytic process. In addition, the buried noise and complex dimensionality of the stock market data make it difficult to learn or re-estimate the ANN parameters [28].

Due to such kinds of bottlenecks, GNP [29,30] and GRA [31] are proposed to solve these problems. Concretely speaking, the efficiency of GNP for generating stock trading rules has been confirmed in our previous studies [32–35]. Also, GRA is originally developed to reduce a large class association rule set for data mining [31]. Unlike strings for solution representation in GA

[a] Correspondence to: Kotaro Hirasawa. E-mail: hirasawa@waseda.jp

* School of Statistics and Management, Shanghai University of Finance and Economics, Shanghai 200433, China

** Graduate School of Information, Production and Systems, Waseda University, 2-7 Hibikino, Wakamatsu-ku, Kitakyushu, Fukuoka 8080135, Japan

and trees in GP, GRA has the ability to express complex events compactly with graph structures, evading black-box issues and exhaustive mathematical properties needed for encoding in other natural inspired algorithms. As a new evolutionary method, GRA is designed especially for complex problems when searching for the optimal solutions. Moreover, in the proposed approach, the correlation coefficient is easily introduced as a measure for the risk and the portfolio considering the risk is optimized easily by the GRA structure, which simply consists of connected nodes and takes less processing time. As GRA has a flexible, compact and robust structure for the portfolio selection, it can focus on asset relationships to capture the risk and return factors to enable quick and liquid investment transactions. Thus, it is shown that GRA is the best choice from many methods.

In this paper, we first apply GRA to the portfolio selection problem. In order to pick up the most efficient portfolio from a large number of brands, the proposed model considers the correlation coefficient between stock brands as strength, which indicates the relation between nodes in GRA. The algorithm evaluates the relationships between stock brands using a specific measure of strength and generates the optimal portfolio in the final generation. The portfolio selected with GRA is further optimized by the stock trading model of GNP. Generally speaking, the contributions of our proposed method are as follows.

- First, in the conventional Markowitz methods, a combination of brands is given, then the distribution ratio of the capital to each brand is determined by considering the return and risk. However, in our method, the best combination of stocks is selected from a large number of brands by considering the correlation coefficients of the brands in GRA. Then, the distribution of the initial capital and the best strategy for buying and selling stocks are determined by GNP.
- Second, although many AI approaches have been applied to this field, most portfolio selection models in the previous literature only consider the distribution property of investment returns; other factors, such as investors' risk preferences and trading strategies, are not taken into account. Unlike others, these important factors are considered in our study. In a sense, our proposed model is an integrated intelligent model.
- Third, the number of stock brands in the best portfolio in the final generation of GRA can be flexibly defined by users as the brands correspond to nodes in the GRA individuals.

The outline of this paper is as follows. Section 1 presents the introduction and literature review. Section 2 describes the proposed GRA approach in general. In Section 3, we explain the application of GRA to our proposed portfolio selection model. Section 4 describes the stock trading strategy of GNP, which is used to verify the GRA approach. Section 5 presents experimental environments, conditions and results using the integrated intelligent model. The trading profits are presented and compared with other methods. Finally, Section 6 concludes this paper.

2. Genetic Relation Algorithm

In this section, the outline of GRA is explained in brief. Basically, GRA is an extension of GP [20] and GNP [29] in terms of gene structures. The original idea is based on the more general representation ability of both directed and undirected graphs. As a new evolutionary computation method, GRA is used to extract the events from a large number of candidates, which shows the best relations with each other in a GRA individual. In the proposed method, a GRA individual is a set of nodes and edges, which are connected with each other as shown in Fig. 1. Based on this concept, a population is composed of a given number of individuals, which will be evolved by genetic operators, such as selection, crossover and mutation. When it is applied to the portfolio selection, GRA is used to select the optimal portfolio out of a huge number of possible stock brands. There are two kinds of gene structures in GRA, i.e. GRA with directed and undirected edges.

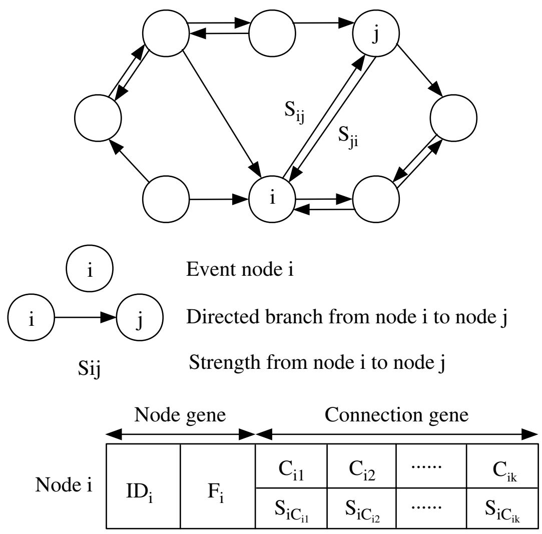

Fig. 1. Basic structure of GRA with directed edges

2.1. GRA with directed edges

Figure 1 shows the basic structure and genotype expression of GRA with directed edges. GRA is composed of nodes and edges, where nodes represent events and directed edges represent the relations between nodes with their strength. As shown in Fig. 1, node i has strength S_{ij} to node j and node j has strength S_{ji} to node i.

Figure 1 also describes the gene of node i, then the set of these genes represents the genotype of GRA individuals. Concretely speaking, ID_i represents an identification number of the node, e.g. $\mathrm{ID}_i = 1$ means node i has the directed edges to other nodes, while $\mathrm{ID}_i = 2$ means node i has the undirected edges to other nodes. F_i denotes the function of node i. In this paper, F_i represents different stock brands in the portfolio. C_{i1}, C_{i2}, ..., C_{ik} show the node number which are connected from node i firstly, secondly and so on. All GRA individuals in a population have the same number of nodes.

Like other evolutionary algorithms, selection, crossover and mutation are used as the genetic operators of GRA. The outline of evolution is described as follows.

(1) Initialize the first population and calculate the fitness of the population.
(2) Generate new individuals for the next generation by tournament selection and genetic operations of crossover and mutation.
(3) Calculate the fitness of the new individuals.
(4) Repeat 2–3 until the terminal condition meets.

The points of GRA can be described as follows: First, GRA extracts appropriate events from a large number of candidates which have the best relations in an individual. Second, all the connections between nodes do not have to be defined, but the connection itself could be evolved. Also, the number of connections between nodes is flexible and it can be defined by users.

2.2. GRA with undirected edges

Figure 2 shows the basic structure of GRA with undirected edges. Like the directed

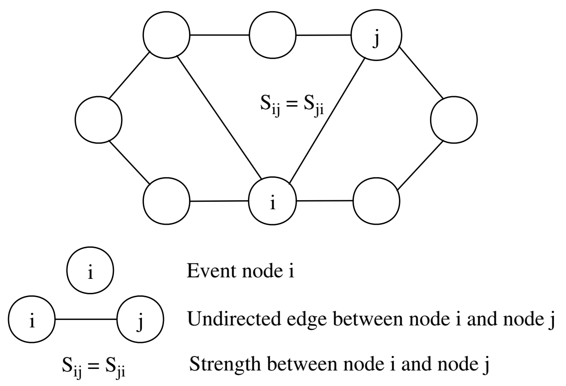

Fig. 2. Basic structure of GRA with undirected edges

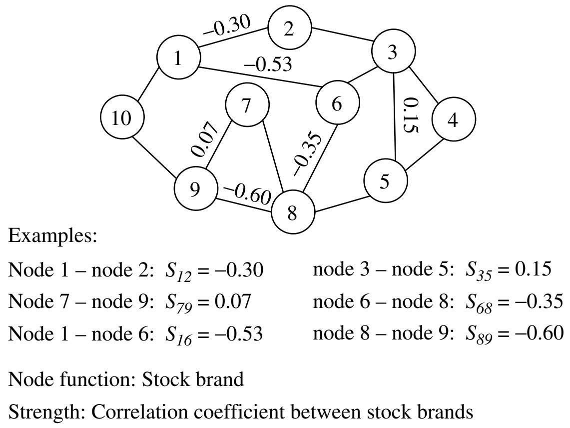

Fig. 3. Genetic relation algorithm for portfolio selection

GRA, the event is also represented by the node, while the relation between nodes is represented by undirected edges with their strength. The relation between node i and node j has a strength of $S_{ij} = S_{ji}$ in undirected GRA, which is different from directed one.

3. Portfolio Selection Using GRA

In our proposed method, GRA with undirected edges are used to construct the portfolio selection model. As shown in Fig. 3, the basic structure of GRA is described as follows: The nodes in GRA are used to represent different stock brands in a portfolio, and the strength between two nodes are used to indicate the relationship between stock brands, i.e. the value of correlation coefficient. The main point of our proposed model is to select a given number of stocks in a portfolio from a large number of brands, so that the correlations among the stock brands in GRA satisfy a certain criterion. In order to maximize the final profit by the buying and selling strategy of GNP [32], we can study what degree of correlation coefficient the stocks should have by GRA method.

3.1. Notations and fitness function of GRA

- D: set of days
- S: set of stock brands
- $S(G)$: set of stock brands in GRA
- $S(G_i)$: set of stock brands whose strength is defined between node i in GRA
- $Price(i,d)$: price of stock brand i on day d
- μ_i: mean of the price of stock brand i
- σ_i^2: variance of the price of stock brand i
- σ_{ij}: covariance between the prices of stock brand i and stock brand j
- ρ_{ij}: correlation coefficient between the prices of stock brand i and stock brand j
- ρ: target value of the correlation coefficient

The object of GRA is to select appropriate $|S(G)|$ stock brands out of a large number of brands $|S|$, which satisfy a certain value of the correlation coefficient, i.e. $-1.0 \leq \rho \leq 1.0$. Therefore, the fitness function of GRA is defined as follow.

$$\text{Fitness} = \frac{1}{|S(G)|} \sum_{i \in S(G)} \frac{1}{|S(G_i)|} \sum_{j \in S(G_i)} (\rho_{ij} - \rho)^2 \quad (1)$$

where

$$\rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j}$$

$$\sigma_i^2 = E[(\text{Price}(i,d) - \mu_i)^2] = \frac{1}{|D|} \sum_{d \in D} (\text{Price}(i,d) - \mu_i)^2$$

$$\sigma_{ij} = E[(\text{Price}(i,d) - \mu_i)(\text{Price}(j,d) - \mu_j)]$$
$$= \frac{1}{|D|} \sum_{d \in D} (\text{Price}(i,d) - \mu_i)(\text{Price}(j,d) - \mu_j)$$

$$\mu_i = E[\text{Price}(i,d)] = \frac{1}{|D|} \sum_{d \in D} \text{Price}(i,d)$$

In the fitness function of (1),

- if ρ is around 1.0, then stock brand i and stock brand j have positive correlation.
- if ρ is around -1.0, then stock brand i and stock brand j have negative correlation.
- if ρ is around 0.0, then stock brand i and stock brand j have no correlation.

The fitness function evaluates the GRA individuals so that the strengths between stock brands have the target value of the correlation coefficient ρ. Generally, according to the portfolio theory, it is preferable to select $|S(G)|$ stock brands which have small correlations. It is our interest to find out the target value of the correlation coefficient ρ in the fitness function. By the portfolio selection model of GRA, the stock brands having large correlations with each other are expected to be eliminated, as they always cause high risk in a portfolio.

3.2. Genetic operators of GRA

In this sub-section, the genetic operators in the evolution phase are introduced. In order to get the best individual, the function of nodes in GRA should be changed, which can be realized effectively by genetic operations. GRA has three kinds of genetic operators: selection, crossover and mutation. In GRA, mutation operation could be executed not only on the connections between nodes but also on the node functions.

3.2.1. *Selection*

In each generation, all of the individuals are ranked by their fitness values and the best individual in the current generation is preserved for the next generation by elite selection. Then, tournament selection of individuals is carried out to reproduce the next generation.

3.2.2. *Crossover*

As shown in Fig. 4, crossover is executed between two parents and two offspring are generated. The procedure of crossover is as follows.

- Select two individuals using tournament selection twice and produce them as parents.
- Each node is selected as a crossover node with the probability of P_c.

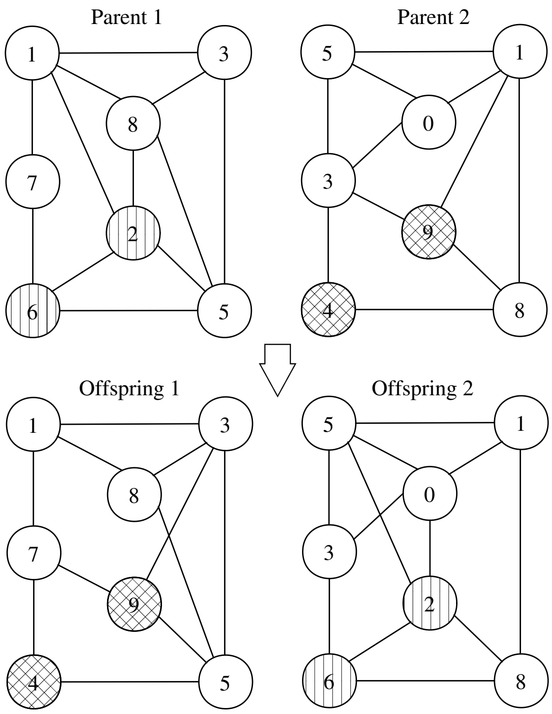

Fig. 4. Crossover

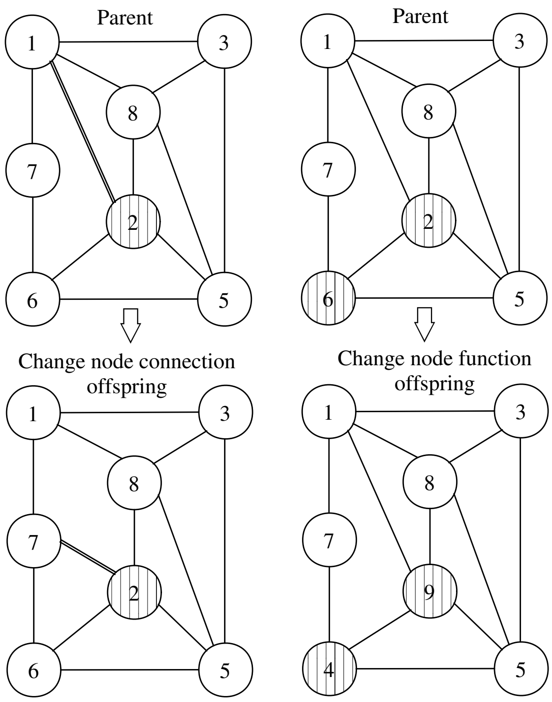

Fig. 5. Mutation

- Two parents exchange the genes of the corresponding crossover nodes.
- Generated new individuals become the new ones of the next generation.

3.2.3. Mutation Figure 5 shows an example of the mutation operator. Mutation is executed in one individual and a new one is generated. The procedure of mutation is as follows.

- Select one individual as a parent using tournament selection.
- Mutation operation

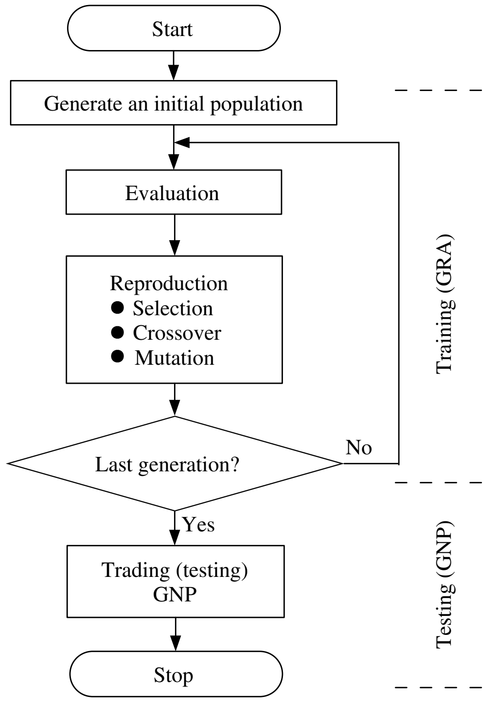

Fig. 6. Flowchart of GRA

— Change connection: Each node branch ($C_{i1}, C_{i2}, \ldots, C_{ik}$) is selected with the probability of P_m, and the selected branch is reconnected to another node.
— Change node function: Each node function (F_i) is selected with the probability of P_m, and the selected function is changed to another one.

- Generated new individual becomes the new one of the next generation.

3.3. Flowchart of GRA Figure 6 shows the flowchart of GRA. For the first GRA population, each individual is generated assigning a certain stock brand selected randomly from a huge number of brands to one of the nodes in GRA. It is ensured that all nodes are different within one individual. Especially, when the same nodes happen to be in the same GRA individual, it will be eliminated to ensure that all stock brands are different in a portfolio. In the next, evaluation of the individuals is carried out according to their fitness values. At the reproduction phase, selection, crossover and mutation are used as genetic operators to generate the population for the next generation. This process is repeated until the last generation. Finally, after obtaining the best individual in the last generation, it is tested by the stock trading model of GNP.

The above procedure can be summarized as follows (Fig. 7):

- Step 1: Evaluate the relationships between stock brands using a measure of correlation coefficients and generate the optimized portfolio in the final generation of GRA, which means the selected portfolio has the lowest value of risk.
- Step 2: Test the portfolio by using the stock trading model of GNP, which considers the technical indices and candlestick chart as trading signals to make decisions. Especially, GNP trading model can distribute the initial capital into different stock brands according to their profitability, and thus maximizes the profits consequently.
- Step 3: Calculate the total profit of the portfolio, and investors can choose the optimal portfolio as their investment decision, i.e. the best combination of stock brands.

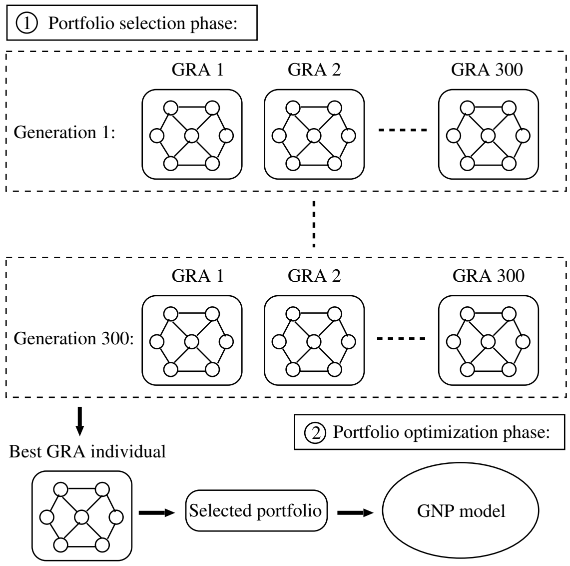

Fig. 7. Procedure of portfolio selection using GRA&GNP integrated model

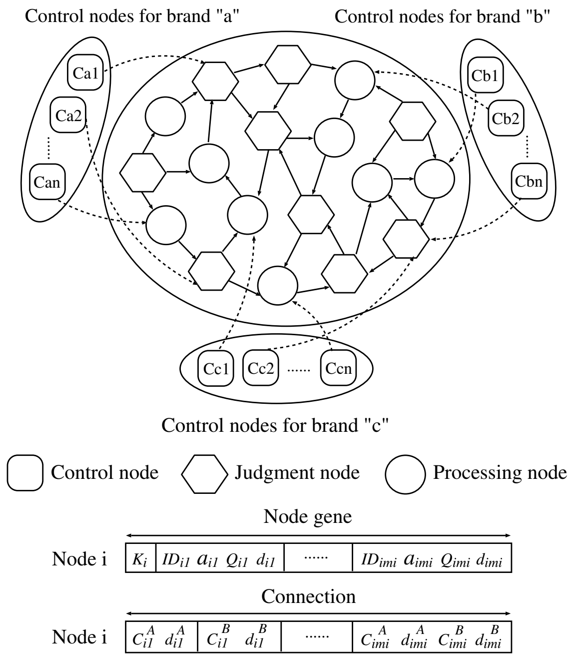

Fig. 8. Basic structure of GNP

4. Stock Trading Strategy of GNP

In this section, an improved stock trading strategy based on GNP [29,32] is applied to optimize the portfolio selected by GRA method. It is a kind of integrated intelligent model by considering both the portfolio selection and trading strategies. The overall procedure for the GNP approach is presented as follows.

- First, the role of GNP is to determine the distribution of the initial capital to each brand selected by GRA, and also determine the time of buying and selling stocks.
- Second, as GNP is evolved with training data, the brands which obtain larger profitability in the training period can have the initial capital more than other brands in the portfolio. As a result, the distribution ratio of the capital and the time of buying and selling stocks are determined by the transition of judgment nodes and processing nodes in GNP.
- Third, as the features of GNP method, the combination of evolution and learning is realized to take the appropriate trading actions, i.e. reinforcement learning is used to select subnode functions in GNP according to their Q values. Moreover, technical indices and candlestick chart are used as judgment functions to get the information from the stock market depending on the situation, thus, the stock trading strategy of GNP can be carried out effectively.

4.1. Basic structure of GNP

GNP is composed of control nodes, judgment nodes and processing nodes, which are connected to each other. Figure 8 shows a basic structure of GNP. Judgment nodes have if-then type branch decision functions, which return judgment results for assigned inputs and determine the next node. Processing nodes take buying and selling actions for the stock trading. Once GNP is booted up, the execution starts from the control node. The genotype expression of GNP node is also shown in Fig. 8. Concretely speaking, K_i represents the node type, and ID_{ip} represents an identification number of the node function at subnode ip. a_{ip} is a parameter which represents the threshold for determining buying or selling stocks in a processing node. Q_{ip} means Q value which is assigned to each state and action pair. C^A_{ip}, C^B_{ip}, ... show the node number of the next node. Judgment node determines the upper suffix of the connection genes depending on the judgment result. d_{ip} is the time delay spent on the judgment or processing, while d^A_{ip}, d^B_{ip}, ... are time delays spent on the node transition from subnode ip to the next node. Generally speaking, GNP uses one of the groups of control nodes for one brand of stocks (as shown in Fig. 8), so that GNP could deal with multi brands in the portfolio selected by GRA. The same as other evolutionary methods, GNP has three kinds of genetic operators: selection, crossover and mutation.

4.2. Judgment function of GNP

When a current activated node i is a judgment node, first, one Q value is selected from $Q_{i1}, \ldots, Q_{im_i}$ based on ϵ-greedy policy (m_i is the number of subnodes in judgment and processing nodes). Then, the corresponding function ID_{ip} is selected. The gene ID_{ip} shows a technical index or a candlestick chart GNP judges at node i. In the GNP stock trading model, technical indices and candlestick chart are used as judgment functions. Concretely speaking, each judgment node uses one of the following technical indices for its judgment: rate of deviation (ROD) from moving average, relative strength index (RSI), rate of change (ROC), volume ratio (VR), rank correlation index (RCI), stochastics, golden/dead cross and moving average convergence and divergence (MACD). When the judgment function of candlestick is selected, GNP refers to the candlestick chart patterns to determine the next node. The candlestick chart is a useful tool to visualize the stock prices so that investors can detect the patterns which can be used to predict future stock price movements. As illustrated in Fig. 9, the candlestick chart consists of a rectangle and two shadow lines. The judgment function of the candlestick chart is executed as follows. When the selected subnode has a judgment function of the candlestick chart, GNP judges the current day's candlestick by the price information. In the GNP stock trading model, we use six typical candlestick chart patterns as shown in Fig. 9.

4.3. Processing function of GNP

When the current node is a processing node, the following will be carried out:

1. First, one Q value is selected from $Q_{i1}, \ldots, Q_{im_i}$ based on ϵ-greedy policy. That is, the maximum Q value among $Q_{i1}, \ldots, Q_{im_i}$ is selected with the probability of $1-\epsilon$, or a random one is selected with the probability of ϵ. Then, the corresponding ID_{ip} is selected.

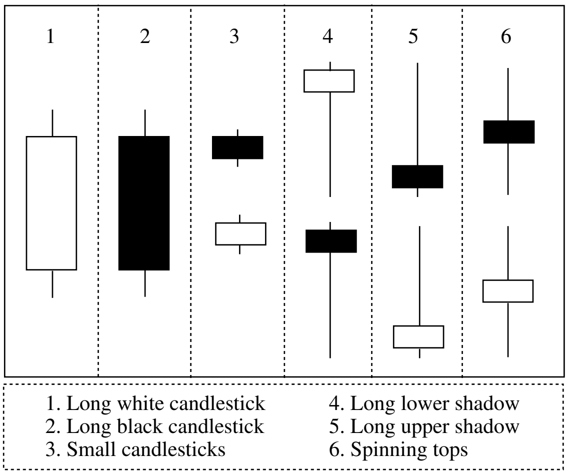

Fig. 9. Candlestick chart patterns

2. Calculate an average value of the IMXs obtained at the judgment nodes executed in the transition from the previous processing node to the current processing node.

$$A_t = \frac{1}{|I'|}\sum_{i' \in I'} \mathrm{IMX}(i')$$

where I' shows a set of suffixes of the judgment nodes executed in the transition from the previous processing node to the current processing node. $\mathrm{IMX}(i')$ shows an IMX output at node $i' \in I'$. However, when a judgment node of the candlestick chart was executed or an IMX output is zero at a judgment node of golden cross, dead cross and MACD, the node number is excluded from I' for calculating A_t.
3. Determine buying or selling:

   - In the case of $\mathrm{ID}_{ip} = 0$ (buy): if $A_t \geq a_{ip}$ and we do not have any stocks, GNP buys the stock of brand $b \in S(G)$ (see Section 3.1) as much as possible using the initial budget of brand b. Otherwise, GNP takes no action.
   - In the case of $\mathrm{ID}_{ip} = 1$ (sell): if $A_t < a_{ip}$ and we have stocks, GNP sells all of the stocks of brand $b \in S(G)$ in hand. Otherwise, GNP takes no action.

4. If ID_{ip} is selected, the next node number becomes C^A_{ip} when the current node is transferred to the next node.

4.4. Initial budget distribution in the portfolio using GNP

In this subsection, how to determine the initial budget of each selected brand in the portfolio is described in the dealing of stocks using GNP. This initial budget optimization system has been constructed by training phase and validation phase (Fig. 10). Especially, the budget for each brand does not change during the validation period, although it changes generation by generation in the training period.

4.4.1. Notations

- Initial*(t)*: initial budget for training
- Initial*(v)*: initial budget for validation
- Initial*(t, b, n)*: initial budget of brand b in the nth generation of training
- Initial*(v, b)*: initial budget of brand b for validation
- T: temperature parameter
- $S(G)$: set of stock brands in GRA
- b: brand index
- N: number of generations

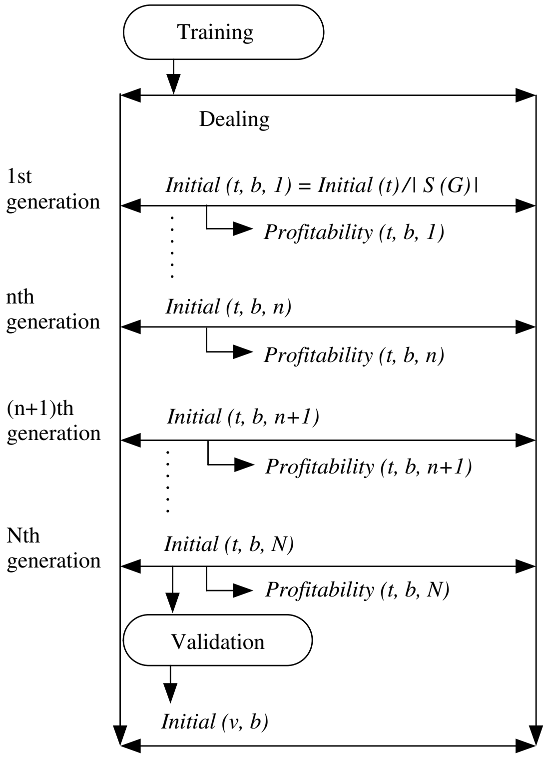

Fig. 10. Training and validation phase

- n: generation number
- Sell*(t, b, n)*: sum of the money obtained by selling the stocks of brand b in the *nth* generation of training
- Buy*(t, b, n)*: sum of the money paid for buying the stocks of brand b in the nth generation of training
- Profit*(t, b, n)*: money gained or lost during the dealing of brand b in the nth generation of training, where Profit*(t, b, n)* = Sell*(t, b, n)* − Buy(t, b, n)
- Profitability(t, b, n): profitability gained during the dealing of brand b in the nth generation of training, where Profitability *(t, b, n)* = $\frac{\text{Profit}(t,\, b,\, n)}{\text{Initial}(t,\, b,\, n)}$
- Profit*(v, b, d)*: money gained or lost during the dealing of brand b until day d for validation

We use GNP to determine the initial budget of each stock brand in the portfolio, where each brand corresponds to each group of control nodes as shown in Fig. 8. In GNP, judgment nodes check the technical indices and candlestick chart patterns, and processing nodes work for buying or selling stocks. One of the groups of control nodes is assigned to each stock brand $b \in S(G)$. The current activated node returns to one of the control nodes of each brand after transiting m processing nodes from the last control node. That is, the current activated node returns to C_{b2} first, and next C_{b3}, $C_{b4}, \ldots, C_{bn}$ [34]. Trading timing is also determined by the GNP evolution.

The fitness function of the GNP in the nth generation is defined as follows,

$$\mathrm{Fitness}(n) = \sum_{b \in S(G)} \mathrm{Profit}(t, b, n).$$

In the following, how to determine the portfolio is explained.

4.4.2. Training phase The initial budget of brand b in the first generation of training phase is calculated by (2).

$$\mathrm{Initial}(t, b, 1) = \frac{\mathrm{Initial}(t)}{|S(G)|} \tag{2}$$

After the first generation, we can get the following profit and profitability.

$$\text{Profit}(t,b,1) = \text{Sell}(t,b,1) - \text{Buy}(t,b,1)$$
$$\text{Profitability}(t,b,1) = \frac{\text{Profit}(t,b,1)}{\text{Initial}(t,b,1)}$$

Then, the initial budget Initial*(t, b, 2)* of brand b in the second generation can be calculated by (3) considering the profitability gained in the first generation.

$$\text{Initial}(t,b,2) = \frac{\exp(\text{Profitability}(t,b,1)/T)}{\sum_{b\in S(G)} \exp(\text{Profitability}(t,b,1)/T)}\,\text{Initial}(t) \quad (3)$$

Likewise, the initial budget Initial $(t,\ b,\ n+1)$ of brand b in the $(n+1)$th generation can be calculated by (4) considering the profitability gained in the nth generation.

$$\text{Initial}(t,b,n+1) = \frac{\exp(\text{Profitability}(t,b,n)/T)}{\sum_{b\in S(G)} \exp(\text{Profitability}(t,b,n)/T)}\,\text{Initial}(t) \quad (4)$$

where

$$\text{Profit}(t,b,n) = \text{Sell}(t,b,n) - \text{Buy}(t,b,n)$$
$$\text{Profitability}(t,b,n) = \frac{\text{Profit}(t,b,n)}{\text{Initial}(t,b,n)}$$

In our previous study, we tested the various values of T in (3), which is called temperature parameter. As we know, according to the Boltzman distribution theory, different T value defines different distributions. When it is applied to this portfolio study, T value indicates the initial budget of each brand distributed by GNP. If $T \to \infty$, GNP distributes the initial budget to each stock evenly. On the other hand, if $T \to 0$, all the money is distributed to the stock brand which obtain the best profitability. It is found in our study that we can get a good fitness value in the training period when T is set to 0.003. Therefore, we set the value of 0.003 for the temperature parameter T in our simulations.

4.4.3. Validation phase The initial budget Initial*(v, b)* of brand b in the validation phase is given as follows.

$$\text{Initial}(v,b) = \frac{\exp(\text{Profitability}(t,b,N)/T)}{\sum_{b\in S(G)} \exp(\text{Profitability}(t,b,N)/T)}\,\text{Initial}(v) \quad (5)$$

where, Initial*(v)* is the initial budget for the validation and N is the number of generations.

GNP individual starts its operation from one of the control nodes in the validation and the activated node is transferred to a judgment node or a processing node. At the processing node, the trading is executed using the opening price of the day. The concrete procedure of the trading is as follows.

- If the current node is a judgment node, it determines the next node depending on the judgment result. If the current node is a processing node, it transits to the next node after buying or selling stocks.
- Calculate the available budget of each brand whenever the buying signal occurs in the processing nodes during the transition of each brand.
- When the processing node is executed m times from the last control node, the next node is determined by the next control node.
- Calculate the profit and profitability of each brand at the end of the trading.

5. Experimental Results

In order to confirm the effectiveness of GRA for the portfolio selection model, we carried out the trading simulations by GNP using the best GRA individual that was obtained in the last generation. The simulation is divided into two steps: one is used for the training of GRA and the other is used for the training and testing of GNP.

- Training (GRA): January 4, 2001 to December 30, 2003
- Training (GNP): January 4, 2001 to December 30, 2003
- Testing (GNP): January 5, 2004 to December 30, 2004

5.1. Performance of GRA

5.1.1. Experimental conditions of GRA Table I shows the parameters of the evolution of GRA. The total number of nodes in each individual of GRA is 10 which indicate 10 different stock brands in a portfolio. Those stock brands are selected from 500 companies listed in the first section of Tokyo stock market in Japan. The content F_i in each node, i.e. the stock brand, is determined randomly at the beginning of the first generation, and changed appropriately by evolution.

The initial connections between nodes are also determined randomly at the first generation. At the end of each generation, 179 new individuals are produced by mutation, 120 new individuals are produced by crossover, and the best individual is preserved. The other parameters for crossover and mutation are the ones showing good results in the simulations. The terminal condition is 300 generations.

5.1.2. Simulation results of GRA Figure 11 shows the average processing time when the number of edges in a GRA individual is changed. It is an example when the target correlation coefficient ρ is set to 0.0. It is clear from Fig. 11 that when the number of edges increases, the average processing time also increases because of the complexity of the network structures.

Figure 12 shows the average fitness values when the number of edges in a GRA individual is changed using the data from 2001 to 2003, and these lines are the average values over 30 independent simulations. From Fig. 12, we can see that the differences of the

Table I. Parameter conditions for evolving GRA

Number of individuals = 300
(mutation: 179, crossover: 120, elite: 1)
Number of generations = 300
Number of nodes = 10
$P_c = 0.3$, $P_m = 0.1$

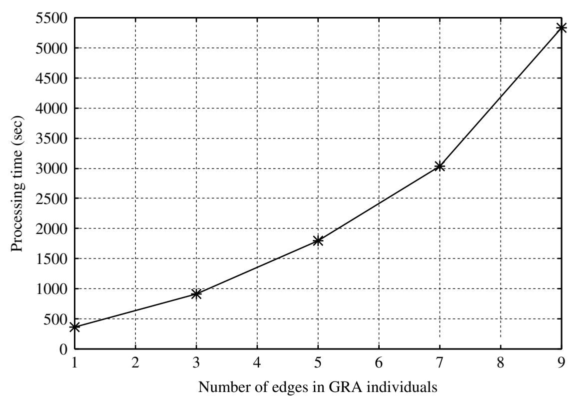

Fig. 11. Processing time when changing the number of edges in GRA

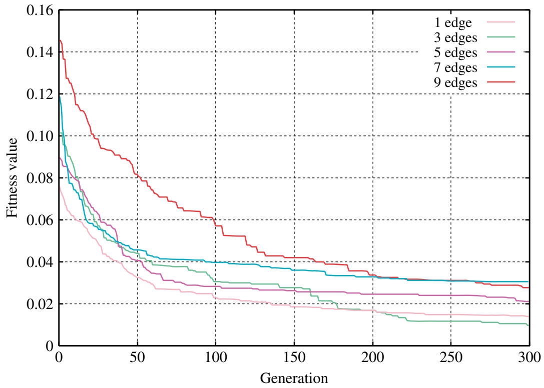

Fig. 12. Average fitness value when changing the number of edges in GRA

Table II. Profit and profitability comparison using different number of branches (yen)

ρ value	1 branch	Full branches
−0.8	2,766,084 (5.53%)	2,631,280 (5.26%)
−0.6	2,672,986 (5.35%)	2,897,513 (5.80%)
−0.4	3,179,176 (6.36%)	3,300,970 (6.60%)
−0.2	3,384,061 (6.77%)	3,261,387 (6.52%)
0.0	3,797,128 (7.59%)	3,673,806 (7.35%)
0.2	3,502,800 (7.01%)	3,607,220 (7.21%)
0.4	2,973,384 (5.95%)	3,180,279 (6.36%)
0.6	3,084,725 (6.17%)	2,862,180 (5.72%)
0.8	2,580,135 (5.16%)	2,500,410 (5.00%)

fitness values between one edge and a large number of edges become small as the generation goes on. This phenomenon occurs because the portfolios with more connection edges have better ability to assess the relational risk among stock brands, which improves the ability to incorporate brands with better risk profiles. However, it is observed that there are no considerable differences among the fitness values with different number of edges in GRA. From Fig. 12, we can also see that more edges in GRA individuals cause slow fitness convergence. The important reason for this phenomenon is that the GRA structures become complex when the number of edges increase, making the processing time increase. This is true when the number of nodes in GRA is around 10, and if the number of nodes is much larger, i.e. there are more stock brands in the portfolio, the number of edges should be increased in order to consider the relational risk among stock brands more sufficiently.

Table II shows the profitability comparison using different number of edges with different value of correlation coefficient ρ, while the initial budgets for training and validation, i.e. Initial(t) and Initial(v) are equal to 50,000,000 Japanese yen. The profitabilities are the simulation results obtained with the integrated intelligent model consisting of GRA and GNP. From Table II, it is found that comparable results are obtained with only one edge compared to full edges of each node in GRA individual. As a small number of edges can save the processing time as shown in Fig. 11, and comparable fitness value and profitability are obtained in general, it is unnecessary to consider the connections of full edges between nodes. Therefore, only one branch is used for each node in the evolution of GRA, which can be evolved by the crossover and mutation.

5.2. Validation by the stock trading model of GNP

5.2.1. Experimental conditions of GNP Table III shows the parameters for the evolution of GNP method. GNP uses the judgment nodes to judge the information from stock markets, and uses the processing node to take buying and selling actions. Five control nodes are assigned to each brand. The total number of nodes in each individual is 80 including 10 processing nodes, 20 judgment nodes and 50 control nodes. The initial connections between nodes are also determined randomly at the first generation. At the end of each generation, new individuals are produced by selection, crossover and mutation. The initial Q values are set at zero.

Table III. Parameter conditions for evolving GNP

Number of individuals = 300
(mutation: 179, crossover: 120, elite: 1)
Number of nodes = 80 (judgement node = 20,
processing node = 10, control node = 50)
Number of sub-node in each node = 2
$P_c = 0.1$, $P_m = 0.03$, $\alpha = 0.1$, $\gamma = 0.3$, $\epsilon = 0.1$

In the validation phase of the stock trading model of GNP, we suppose that the initial funds are 50,000,000 Japanese yen in both training and testing periods. Especially, when we use GNP to test the best portfolio generated by GRA, one GNP individual has 10 groups of control nodes, each of which deals with one brand, so one GNP can deal with 10 brands in the portfolio simultaneously.

5.2.2. Simulation results of GNP In order to test the proposed method, Table IV shows the comparison of profits with different target values of the correlation coefficient. Concretely speaking, in the GRA&GNP integrated model, we carry out the simulations by setting the various values of ρ in the fitness function of (1). The value of ρ indicates positive or negative correlation between different stock brands, and it has strong effects to the portfolio selection. Table IV presents the average profits of the portfolio selected by GRA over 30 independent simulations when the correlation coefficient ρ is set at different values. From the results, it is clarified that we can get a good profit in the testing period when ρ is set to 0.0. Therefore, we set the value of 0.0 for the parameter ρ in our simulations.

From Tables V–VII, we compare the proposed method GRA&GNP with many other intelligent methods using neural network (NN) [27] and GA [12]. Especially, we adopt three forecasting models in the NN method, i.e. random walk (RW) model, adaptive exponential smoothing (AES) model and autoregressive integrated moving average (ARIMA) model to predict these financial series. To test the effectiveness of GRA&GNP method, different data periods are examined in our experiments as follows.

- Period 1: (Training: January 4, 1999 to December 28, 2001; Testing: January 4, 2002 to December 30, 2002)
- Period 2: (Training: January 4, 2000 to December 30, 2002; Testing: January 6, 2003 to December 30, 2003)
- Period 3: (Training: January 4, 2001 to December 30, 2003; Testing: January 5, 2004 to December 30, 2004)

The simulation results in Tables V–VII show the proportions of different stock brands and the portfolio return using different models. From the experimental results, it is clarified that the proposed GRA&GNP model outperforms other models with the data of different periods.

Figure 13 shows the fitness and profit curves in period 3, i.e. the best Fitness(n) and Profit(t, b, n) of each brand $b \in S(G)$ in the nth generation using the data from 2001 to 2003, and the line is the average value over 30 independent simulations. From Fig. 13, we can see that GNP with reinforcement learning can obtain larger profits for the training data as the generation goes on.

Figure 14 shows the profits change Profit(v, b, d) of the selected 10 brands in the testing of period 3 by GNP model, i.e. using

Table IV. Comparison of profits with different target values of the correlation coefficient (profit [yen])

ρ	−0.02	−0.04	−0.06	−0.1	−0.2	−0.4	−0.6	−0.8
GRA&GNP	3,501,807	3,251,973	3,163,410	3,352,168	3,384,061	3,179,176	2,672,986	2,766,084
ρ	0.0							
GRA&GNP	3,797,128							
ρ	0.02	0.04	0.06	0.1	0.2	0.4	0.6	0.8
GRA&GNP	3,451,030	3,600,963	3,180,345	3,275,730	3,102,800	2,973,384	3,084,725	2,580,135

Table V. Proportions of different stock brands in the portfolio using different models (period 1)

	GRA&GNP	RW-NN	AES-NN	ARIMA-NN	GA
Brand	0.1736	0.2014	0.0180	0.4075	0.0053
Brand	0.0013	0.0000	0.0056	0.0000	0.1357
Brand	0.0116	0.1783	0.3622	0.0047	0.0363
Brand	0.2027	0.0065	0.0000	0.1849	0.0758
Brand	0.0038	0.0853	0.2521	0.0523	0.5701
Brand	0.0154	0.1071	0.0000	0.2016	0.0000
Brand	0.4016	0.0023	0.1048	0.0039	0.0000
Brand	0.0079	0.0040	0.1237	0.0000	0.1142
Brand	0.0800	0.3227	0.0029	0.0026	0.0000
Brand	0.1021	0.0924	0.1307	0.1425	0.0626
Portfolio return (%)	6.18	4.29	5.20	2.32	3.73

Table VI. Proportions of different stock brands in the portfolio using different models (period 2)

	GRA&GNP	RW-NN	AES-NN	ARIMA-NN	GA
Brand	0.0734	0.0355	0.2671	0.0000	0.069
Brand	0.0031	0.1736	0.0393	0.0071	0.0022
Brand	0.1653	0.0082	0.0000	0.1483	0.0000
Brand	0.0085	0.0000	0.0062	0.2673	0.0000
Brand	0.1032	0.1957	0.0328	0.1069	0.0385
Brand	0.3507	0.2294	0.0736	0.0522	0.0052
Brand	0.2071	0.0926	0.1752	0.3061	0.5002
Brand	0.0157	0.2458	0.0924	0.0025	0.0063
Brand	0.0028	0.0000	0.1017	0.1096	0.2759
Brand	0.0702	0.0192	0.2117	0.0000	0.1027
Portfolio return (%)	5.95	5.37	3.01	4.12	2.18

Table VII. Proportions of different stock brands in the portfolio using different models (period 3)

	GRA&GNP	RW-NN	AES-NN	ARIMA-NN	GA
Brand	0.0892	0.1528	0.3759	0.0527	0.1675
Brand	0.0681	0.0000	0.0000	0.0091	0.0381
Brand	0.3092	0.2415	0.0376	0.1024	0.0690
Brand	0.0913	0.0037	0.1653	0.0000	0.0000
Brand	0.0427	0.0325	0.0762	0.2740	0.0735
Brand	0.2198	0.1322	0.1729	0.0970	0.4300
Brand	0.0443	0.1526	0.0000	0.2524	0.0000
Brand	0.0536	0.0756	0.0030	0.0287	0.1762
Brand	0.0087	0.1216	0.0533	0.1163	0.0000
Brand	0.0731	0.0875	0.1158	0.0674	0.0457
Portfolio return (%)	7.59	5.31	5.92	3.18	3.25

the best portfolio obtained with GRA method when the value of parameter ρ is set to 0.0. We carried out the dealing of these 10 brands using the data of 2004. From Fig. 14, we can see that the profit keep increasing during the testing period. The advantage of the proposed integrated intelligent model is that it can optimize not only the portfolio selection, but also optimize the distribution of initial budget to each brand in the portfolio automatically, and the brands which obtain larger profitability can have the initial budget more than other brands. As a result, with this efficient portfolio optimization system, we can obtain much profits in the trading of those brands.

6. Conclusions

In this paper, we applied GRA and GNP to the portfolio selection problem. In order to pick up a fixed number of the most efficient brands, the algorithm evaluates the relationships between stock brands using a specific measure of strength, i.e. correlation coefficient between stocks, and generate the optimal portfolio in the final generation. We carried out the experiments selecting 10 brands out of 500 companies listed in the first section of Tokyo stock market in Japan for 4 years. In the experiments, the efficiency of GRA method is confirmed by the stock trading model of GNP that has been proposed in our previous study. Compared

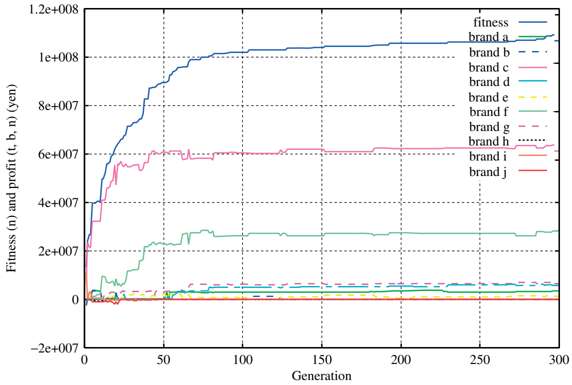

Fig. 13. Fitness and profit curves of 10 brands in the training of period 3 by GNP

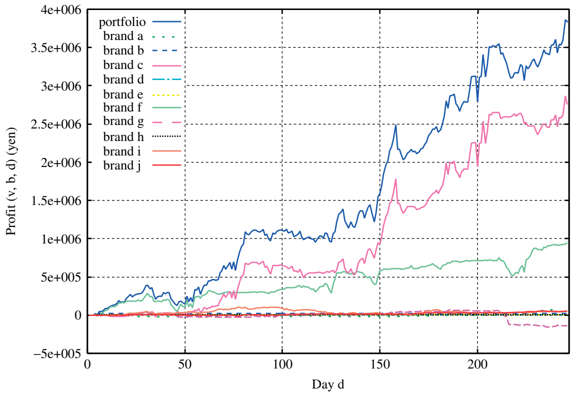

Fig. 14. Profits change of selected 10 brands in the testing of period 3 by GNP

to conventional methods, the proposed integrated intelligent model has two advantages, first, GRA method considers the correlation coefficient as the strength between stock brands to optimize the portfolio, which is different from the conventional methods that select stocks randomly for trading. From the results, it is clarified that we can obtain much profits in the trading of those brands. Second, GNP can adjust the distribution of the initial budget to each selected brand in the portfolio at each generation, and more budget is assigned to the brands with larger profitability. It is different from the Markowitz method in terms that the distribution ratio of the capital to each brand is determined for maximizing the total profits as the risk factor is already considered in GRA. Thus, we can create an effective portfolio optimization system and get more profits by the combination of GRA and GNP.

There remain some further studies in the future. They include, but not restricted to:

- First, the algorithm presented can be further improved by incorporating several new methods to allow the evolution of better GRA individuals.
- Second, the portfolio selection model can be enhanced by modifying the fitness function, i.e. we will study how to take into account traders' preference for the brands in the fitness function of GRA. Also, it is necessary to study various measures in addition to the correlation coefficient.
- Third, the system will be tested on many kinds of markets and with different data periods.

Moreover, the system will incorporate additional financial analysis tools to increase the information content from the market.

Acknowledgments

This research is supported by National Natural Science Foundation of China (Grant No. 71101083); Innovation Program of Shanghai Municipal Education Commission (Grant No. 12ZZ072); Shanghai Pujiang Program supported by Shanghai Science and Technology Commission (Grant No. 11PJ1403300); "Chen Guang" project supported by Shanghai Municipal Education Commission and Shanghai Education Development Foundation (Grant No. 11CG37); the Fundamental Research Funds for the Central Universities for Shanghai University of Finance and Economics; Leading Academic Discipline Program, 211 Project for Shanghai University of Finance and Economics, Project Number 2113 and Shanghai Leading Academic Discipline Program, Project Number B803.

Appendix: 10 Selected Stocks in the Portfolio

The list of selected 10 companies in Figs 13 and 14 is as follows.

(a) Nissin Foods Products Co., Ltd.
(b) Hitachi Chemical Co., Ltd.
(c) ToTo Ltd.
(d) Toshiba Corporation
(e) Honda Motor Co., Ltd.
(f) Yamaha Corporation
(g) Toyota Tsusho Corporation
(h) All Nippon Airways Co., Ltd.
(i) Chubu Electric Power Co., Inc.
(j) Toho Co., Ltd.

References

(1) Markowitz HM. Portfolio selection. *Journal of Finance* 1952; **7**:77–91.
(2) Markowitz HM. The optimization of a quadratic function subject to linear constraints. *Naval Research Logistics Quarterly* 1956; **3**:111–133.
(3) Markowitz HM. *Portfolio Selection: Efficient Diversification of Investments*. Wiley, Yale University Press: New York; 1959.
(4) Markowitz HM. *Mean-Variance Analysis in Portfolio Choice and Capital Markets*. Blackwell Publishers: Oxford; 1987.
(5) Best MJ, Kale JK. Quadratic programming for large-scale portfolio optimization. In *Financial Services Information Systems*. Keyes J (ed). CRC Press LLC: Boca Raton; 2000; 513–529.
(6) Stein M, Branke J, Schmeck H. Efficient implementation of an active set algorithm for large-scale portfolio selection. *Computers & Operations Research* 2008; **35**(**12**):3945–3961.
(7) Box GEP, Jenkins GM. *Time Series Analysis: Forecasting and Control*. Holden-DaySan: Fransisco, CA; 1976.
(8) Tong H, Lim KS. Threshold autoregressive, limit cycles and cyclical data. *Journal of the Royal Statistical Society Series B* 1980; **42**(**3**):245–292.
(9) Sarantis N. Nonlinearities, cyclical behavior and predictability in stock markets: international evidence. *International Journal of Forecasting* 2001; **17**(**3**):459–482.
(10) Engle RF. Autoregressive conditional heteroskedasticity with estimates of the variance of UK inflation. *Econometrica* 1982; **50**:987–1008.
(11) Fernandez A, Gomez S. Portfolio selection using neural networks. *Computers & Operations Research* 2007; **34**(**4**):1177–1191.
(12) Xia Y, Liu B, Wang S, Lai K. A model for portfolio selection with order of expected returns. *Computers & Operations Research* 2000; **27**(**5**):409–422.
(13) Holland JH. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. University of Michigan Press: Ann Arbor, Michigan; 1975.
(14) Goldberg DE. *Genetic Algorithm in Search, Optimization and Machine Learning*. Addison-Wesley: Reading, Massachusetts; 1989.
(15) Loraschi A, Tettamanzi A, Tomassini M, Verda P. Distributed genetic algorithms with an application to portfolio selection problems. Proceedings of the Artificial Neural Networks and Genetic Algorithms.

Pearson DW, Steele NC, Albrecht RF (eds). Springer: Wein, 1995; 384–387.
(16) Skolpadungket P, Dahal K, Harnpornchai N. Portfolio optmization using multi-objective genetic algorithms. Proceedings of the IEEE Congress on Evolutionary Computation, 2007; 516–523.
(17) Lin CC, Liu YT. Genetic algorithms for portfolio selection problems with minimum transaction lots. *European Journal of Operational Research* 2008; **185**(**1**):393–404.
(18) Lin D, Wang S, Yan H. A multiobjective genetic algorithm for portfolio selection problem. Proceedings of the ICOTA 2001, Hong Kong, 15–17 December 2001.
(19) Oh KJ, Kim TY, Min SH, Lee HY. Portfolio algorithm based on portfolio beta using genetic algorithm. *Expert Systems with Applications* 2006; **30**(**3**):527–534.
(20) Koza JR. *Genetic Programming, on the Programming of Computers by Means of Natural Selection*. MIT Press: Cambridge, MA; 1992.
(21) Potvin J, Soriano P, Vallee M. Generating trading rules on the stock markets with genetic programming. *Computers & Operations Research* 2004; **31**(**7**):1033–1047.
(22) Dempster MAH, Jones CM. A real-time adaptive trading system using genetic programming. *Quantitative Finance* 2001; **1**:397–413.
(23) Marney JP, Fyfe C, Tarbert H, Miller D. Risk adjusted returns to technical trading rules: a genetic programming approach. Computing in Economics and Finance, Society for Computational Economics, Yale University, USA, June, 2001, Paper no. 147, 2001.
(24) Etemadi H, Rostamy AAA, Dehkordi HF. A genetic programming model for bankruptcy prediction: Empirical evidence from Iran. *Expert Systems with Applications* 2009; **36**(**2**):3199–3207.
(25) Enke D, Thawornwong S. The use of data mining and neural networks for forecasting stock market returns. *Expert Systems with Applications* 2005; **29**(**4**):927–940.
(26) Lam M. Neural network techniques for financial performance prediction: integrating fundamental and technical analysis. *Decision Support Systems* 2004; **37**(**4**):567–581.
(27) Yu L, Wang S, Lai KK. Neural network-based mean-variance-skewness model for portfolio selection. *Computers & Operations Research* 2008; **35**(**1**):34–46.
(28) Kim K-J, Han I. Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert Systems with Applications* 2000; **19**(**2**):125–132.
(29) Mabu S, Hirasawa K, Hu J. A graph-based evolutionary algorithm: genetic network programming and its extension using reinforcement learning. *Evolutionary Computation* 2007; **15**(**3**):369–398.
(30) Hirasawa K, Eguchi T, Zhou J, Yu L, Hu J, Markon S. A double-deck elevator group supervisory control system using genetic network programming. *IEEE Transactions on Systems, Man and Cybernetics C* 2008; **38**(**4**):535–550.
(31) Gonzales E, Taboada K, Shimada K, Mabu S, Hirasawa K. Evaluating class association rules using genetic relation programming. Proceedings of the IEEE Congress on Evolutionary Computation 2008, 2008; 731–736.
(32) Chen Y, Mabu S, Shimada K, Hirasawa K. Trading rules on stock markets using genetic network programming with Sarsa learning. *Journal of Advanced Computational Intelligence and Intelligent Informatics* 2008; **12**(**4**):383–392.
(33) Chen Y, Mabu S, Shimada K, Hirasawa K. Real time updating genetic network programming for adapting to the change of stock prices. *IEEJ Transaction on EIS* 2009; **129**(**2**):344–354.
(34) Ohkawa E, Chen Y, Bao Z, Mabu S, Shimada K, Hirasawa K. Buying and selling stocks of multi brands using genetic network programming with control nodes. *IEEJ Transaction on EIS* 2008; **128**(**12**):1811–1819.
(35) Mabu S, Izumi Y, Hirasawa K, Furuzuki T. Trading rules on stock markets using genetic network programming with candle chart. *SICE Transaction* 2007; **43**(**4**):317–322.

Yan Chen (Non-member) received the B.B.A. degree from Sichuan Normal University, China in 2005, and double M.E. degrees from Waseda University, Japan in 2007 and Sichuan University, China in 2008, respectively. In 2010, she received the Ph.D. degree in Engineering from Waseda University, Japan. From April 2010 to August 2010, she was a Postdoctoral Research Fellow in Japan Society for the Promotion of Science. Since September 2010, she has been an assistant professor in School of Statistics and Management, Shanghai University of Finance and Economics.

Kotaro Hirasawa (Member) received the B.E. and M.E. degree from Kyushu University, Japan in 1964 and 1966, respectively. From 1966 to 1992, he was with Hitachi Ltd., where he served as a vice president of Hitachi Research Laboratory. From December 1992 to August 2002, he was a Professor in Graduate School of Information Science and Electrical Engineering of Kyushu University. Since September 2002, he has been a Professor in Graduate School of Information, Production and Systems of Waseda University. Dr Hirasawa is a member of the Society of Instrument and Control Engineers and IEEE.