FuzzyTree Crossover for Multi-Valued Stock Valuation

Pin-Chen (P.C.) Lin
(Corresponding author)
Department of Information Management
Van Nung Institute of Technology
Jungli, Taiwan 320, R.O.C
e-mail: lety@cc.vit.edu.tw

Jiah-Shing Chen
Dept. of Information Management,
National Central University,
Jungli, Taiwan 320, R.O.C.
e-mail: jschen@mgt.ncu.edu.tw

Abstract

Stock valuation is very important for fundamental investors to select undervalue stocks to earn excess profit. However, it may be difficult to use stock valuation results because different models generate different estimates on the same stock. This suggests that the value of a stock should be multi-valued rather than single-valued. We therefore develop a multi-valued stock valuation model based on fuzzy genetic programming. In our fuzzy GP model, the value of a stock is represented as a fuzzy expression tree whose terminal nodes are allowed to be fuzzy numbers. There is little literature available on the crossover operator for our fuzzy trees except the vanilla subtree crossover. This study generalizes the subtree crossover to design a new crossover operator for the fuzzy trees. Since the stock value is estimated by a fuzzy expression tree which calculates to a fuzzy number, the stock value becomes multi-valued. In addition, the resulting fuzzy stock value induces a natural trading strategy which can readily be executed and evaluated. Experimental results indicate that the FuzzyTree crossover is more effective than subtree crossover in terms of expression tree complexity and run time. Second, shorter training periods produce better ROI. It indicates long-term financial statement may distort the intrinsic value of a stock. Finally, the return of multi-valued fuzzy trading strategy is better than that of single-valued and Buy-and-Hold strategy. We suggest that more attention should be put on the multi-valued stock valuation approach.

Keywords: Stock Valuation, Intrinsic Value, Multi-Value, Fuzzy Number, Genetic Programming.

1 Introduction

Stock valuation is the activities of estimating intrinsic value of a business entity. It is important to securities analysis, loan decision, and leveraged buyout analysis and so on. The investors could suffer vast loss if they made improper decisions based on wrong business value. To evaluate a business value, it is necessary to understand the activities disclosed in its financial statements. Conventional methods to stock valuation using financial statements are divided into 3 categories: Asset Appraisal [11], Discounted Present Value [1][13] and Multiples Price [5]. In these valuation methods, using different critical input variables produces different outcome even on the same stock. It implies that the value of a business may be multi-valued rather than single-valued. Another drawback is using well-known functions to design valuation models, which tries to estimate a business value from linear functions under specific assumptions and limitations. It always fails to fully capture flexibility and uncertainty.

On the other hand, the various soft computing technologies provide alternative solutions to financial problems. For example, Fuzzy Logic is used as possibility distribution of portfolios [7][18][19], or credit analysis of loan [8]. Neural Networks are used to predict financial distress [2][3][6]. One of evolutionary computations technique, Genetic Programming (GP) is applied to stock trading market [10], future or option pricing [9], and foreign exchange market [4]. However, few studies used soft computing methods to stock valuation. In this article, we apply both fuzzy numbers to manifest multi-valued uncertainty and Genetic Programming to optimize an effective stock valuation model.

It is known that crossover and selection operators mainly contribute to generate solutions in GP [16]. Subtree crossover operator usually destroy building block (i.e. effective partial trees) because of randomly and blindly choosing crossover points. Hence, many investigators propose new crossover methods to obtain more effective building blocks by reserving crucial schemata. For example, Hierarchical crossover combined Simulated Annealing and Hill Climbing to find correct solutions via shrinking, growth or internal substitution while preserving syntactic correctness [22]. Depth-dependent crossover accumulates building blocks according to the depth of a node. The depth selection ratio is higher for node closer to a root node [20]. Directing crossover reduced the amount of unviable code (bloat) in individuals while searching for a parsimonious solution [21]. It involves the identification of highly fit nodes to use as crossover points during operator application. Island model crossover applies subtree crossover to aborigines and depth-dependent crossover to immigrants with their ages, which demonstrate how long they survive in the demes [17]. It can integrate many schemata to forming a bigger building block of different demes. Dynamic page based crossover was described in terms of a number of pages of all individuals. Pages are expressed a number of instructions, which is dynamic change for all individuals in the population. It evolves succinct solution without penalizing optimization ability [14][15]. These crossover operators only exchange constant schemata of all individuals in population. It derives that no new genotype is generated even swapping partial trees in the dedicated population. Besides, little literature is available on exploring new crossover operator in Fuzzy Genetic Programming except subtree.

The objective of the present study was to develop a FuzzyTree crossover for Multi-Valued stock valuation model which improves convergence phenomenon. We generalize crisp expression trees evolved by GP to fuzzy ones by introducing fuzzy numbers and fuzzy arithmetic operators in the trees. FuzzyTree uses subtree crossover operator if selected crossover point is an internal node; otherwise, the selected terminal nodes would be snipped into pieces and interchanged with each other. It could improve convergence via protecting building block and increasing variety genotypes. Since the stock value is estimated by a fuzzy expression tree which calculates to a fuzzy number, the real stock values becomes Multi-Valued. In addition, the resulting trapezoidal fuzzy stock value induces a natural trading strategy which can readily be executed and evaluated.

2 Method

In our FuzzyTree model, it integrates subtree to produce next-generation fuzzy GP individual. Fuzzy GP individual is represented by a fuzzy GP tree shown in Figure 1, which contains terminal (n_i or v_i) nodes and int nodes (f_i). Each terminal node is represented by a trapezoidal fuzzy number. The detail FuzzyTree crossover algorithm is described in Section 2.1. Section 2.2 introduces the encoding of each fuzzy node. The related arithmetic processing of our fuzzy GP tree is shown in Section 2.3. Evaluating a goodness individual for survive relies on fitness function illustrated in Section 2.4. Finally, we propose a fuzzy trading strategy to obtain better investment returns in Section 2.5.

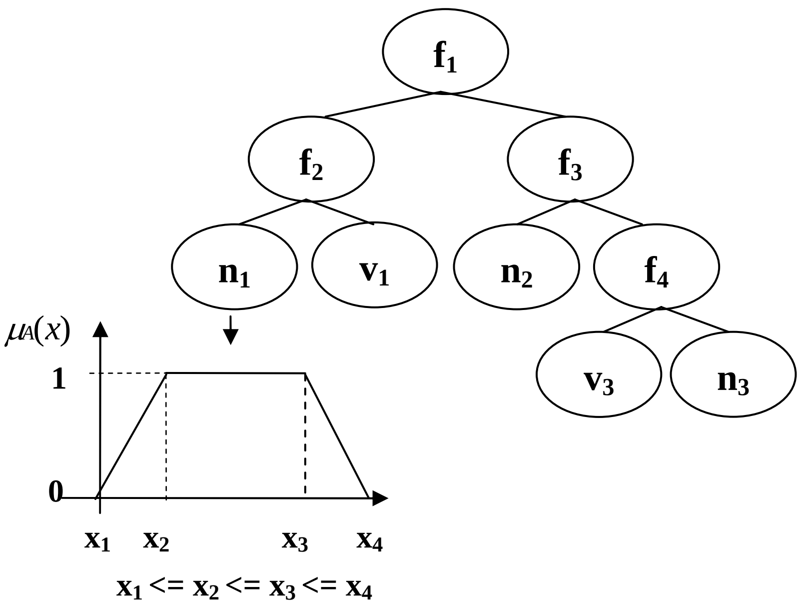


Figure 1: Fuzzy GP tree

2.1 FuzzyTree Crossover

A basic evolutionary algorithm introduces a simple crossover-mutation-evaluation-selection loop as outlined below (Figure 2):

Initialize population;
Evaluate population;
Do Terminal Criteria ≠ true
    Crossover;
    Mutation;
    Evaluate;
    Selection;
End DO
Report the best solution found

Figure 2: Evolutionary algorithm

As a general framework of our proposed FuzzyTree crossover algorithms (Figure 3), $Tree_1$ and $Tree_2$ denote the selected trees from selection method, and $NewTree_1$ and $NewTree_2$ are the generated offsprings by this crossover function. The Selection process is to select relative good solutions and eliminate those not-so-good solutions from parent population. In our model, we use a well-known tournament selection methodology to pick up relative good offsprings, because it achieves better performance [10][17][21]. n_1 and n_2 are random selected crossover points from $Tree_1$ and $Tree_2$, respectively. If both n_1 and n_2 belong to terminal nodes, our proposed FuzzyTree(.) crossover method is performed, otherwise, the conventional GP crossover method Subtree(.) is used.

Input: $Tree_1$, $Tree_2$, Rate_C
Output: $NewTree_1$, $NewTree_2$
    IF rnd < Rate_C **THEN**   // rnd is a random generated number.
        n_1 = TournamentSelection CrossoverPoint ($Tree_1$);
        n_2 = TournamentSelection CrossoverPoint ($Tree_2$);
        IF $n_1 \in$ terminal node and $n_2 \in$ terminal node **THEN**
            FuzzyTree($Tree_1$, $Tree_2$, n_1,n_2, &$NewTree_1$, &$NewTree_2$);
        ELSE
            Subtree(($Tree_1$, $Tree_2$, n_1,n_2, &$NewTree_1$, &$NewTree_2$);
        End IF
    ELSE
        $NewTree_1 \leftarrow Tree_1$;

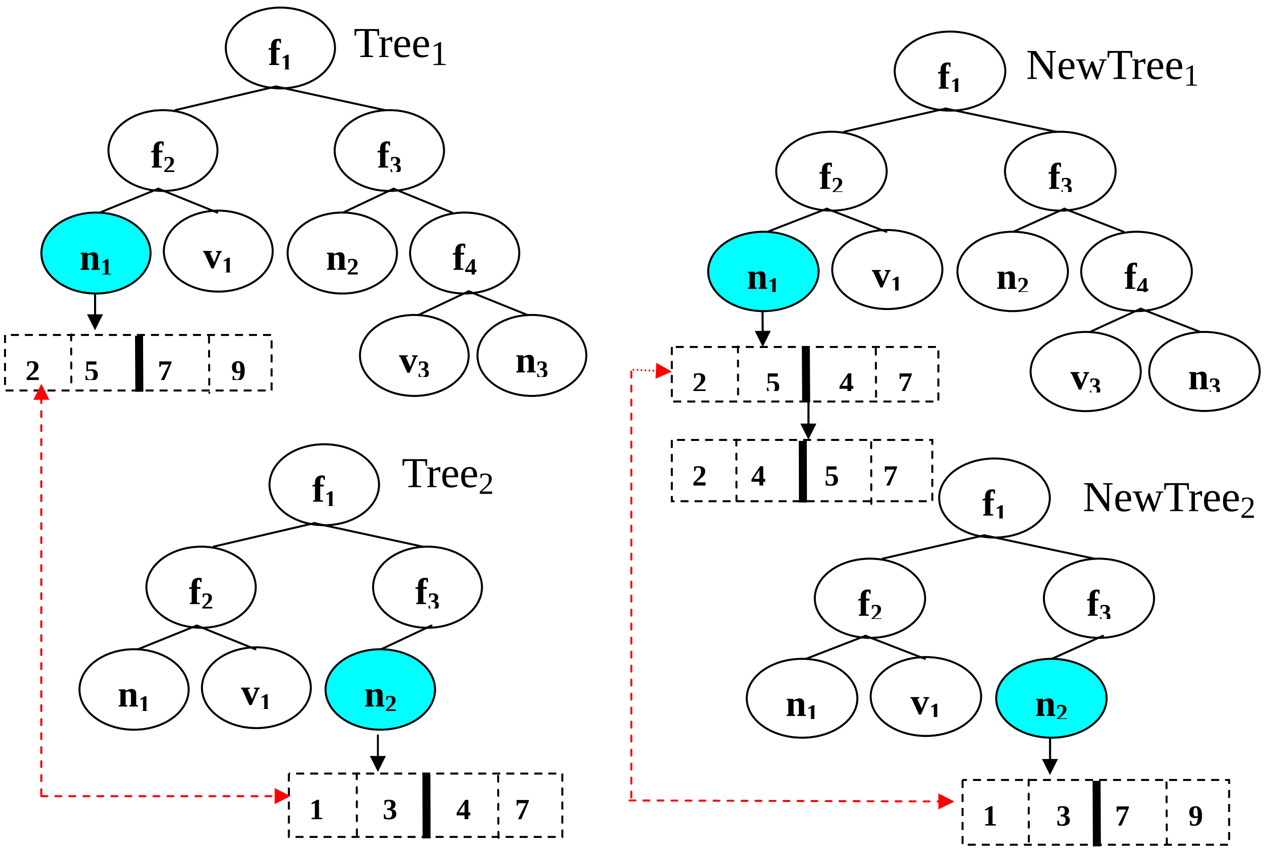
4.13 NewTree$_2$ $\leftarrow$ Tree$_2$;

4.14 **End IF**

Figure 3: FuzzyTree crossover algorithm

- **FuzzyTree(.) Crossover Function**

In this section, FuzzyTree(.) function is performed only when both of terminal nodes (n_1 and n_2) are selected to be crossover points, simultaneously, shown in Figure 4. It means that both n_1 and n_2 should be represented by trapezoidal fuzzy numbers, $x_1 \le x_2 \le x_3 \le x_4$. For example, n_1 = [2, 5, 7, 9] and n_2 = [1, 3, 4, 7] before crossover operator in Figure 4(a). Both of n_1 and n_2 should be snipped into two pieces and interchange the pieces with each other, where the snipped point is random selected. Assume the new generated fuzzy node is $[x_1', x_2', x_3', x_4']$. For maintaining the order of new generated fuzzy numbers, they are sorted increasingly to $[x_1'', x_2'', x_3'', x_4'']$, where $x_1'' \le x_2'' \le x_3'' \le x_4''$. Assume the snipped point is between x_2 and x_3, our crossover operator produces [2, 5, 4, 7] and [1, 3, 7, 9] after interchanging. Then, these two produced fuzzy numbers should be sorted increasingly to [2, 4, 5, 7] and [1, 3, 7, 9], shown in Figure 4(b). Finally, the new generated offsprings NewTree$_1$ and NewTree$_2$ are obtained.



(a) Before crossover (b) After crossover

Figure 4: FuzzyTree crossover

2.2 Encoding

Each individual in a GP population is a fuzzy expression tree, which represents a valuation model. An expression tree consists of terminal nodes and internal nodes. A terminal node can be a financial variable or a constant, while an internal node can be an allowed fuzzy arithmetic operator. The expression tree is shown in Figure 1, where v_1, v_2, ⋯, $v_m \in \{R_1, R_2, \cdots, R_{39}\}$ are financial variables, f_1, f_2, ⋯, $f_k \in \{+, -, \times\}$ are fuzzy operators, and n_1, n_2, ⋯, n_i are trapezoidal fuzzy numbers (constants). There are 39 available financial variables used in our study listed in Table 1. A trapezoidal fuzzy number (constant) can be denoted as a 4-tuple $[x_1, x_2, x_3, x_4]$, as depicted in Figure 1. Each financial variable is a non-fuzzy number (exact value) which is represented as a degenerated trapezoidal fuzzy number with $x_1 = x_2 = x_3 = x_4$.

Table 1. Some of the financial variables

Ratio	Description
R_1	Return on Total Assets (%)
R_2	Current Liabilities (%)
R_3	Earning per Share
…	…
R_{39}	Operation Income Per Employee

2.3 Fuzzy arithmetic

To evaluate an expression tree with trapezoidal fuzzy numbers as terminal nodes, we define several fuzzy arithmetic operations on trapezoidal fuzzy numbers so that the resulting fuzzy numbers will also be trapezoidal. Currently, the supported fuzzy operations in our model are +, - and ×. Let $X = [x_1, x_2, x_3, x_4]$, $Y = [y_1, y_2, y_3, y_4]$ be two trapezoidal fuzzy operands. We define fuzzy +, -, × as follow:

$$X + Y \equiv [x_1+y_1, x_2+y_2, x_3+y_3, x_4+y_4] \quad (1)$$

$$X - Y \equiv [x_1-y_4, x_2-y_3, x_3-y_2, x_4-y_1] \quad (2)$$

$$\begin{aligned} X \times Y \equiv [&\min (x_1y_1, x_4+y_1, x_1+y_4, x_4+y_4), \\ &\min (x_2y_2, x_2+y_3, x_3+y_2, x_3+y_3), \\ &\max (x_2y_2, x_2+y_3, x_3+y_2, x_3+y_3), \\ &\max (x_1y_1, x_4+y_1, x_1+y_4, x_4+y_4)] \end{aligned} \quad (3)$$

It is not difficult to see that the above definitions result in well-formed trapezoidal fuzzy numbers, i.e., $z_1 \leqq z_2 \leqq z_3 \leqq z_4$. Since every operator produces a trapezoidal fuzzy number, an expression tree also yields a trapezoidal fuzzy number, $[z_1, z_2, z_3, z_4]$, which represents the fuzzy value of a stock under consideration. In addition, the trapezoidal fuzzy stock value also induces a natural trading rule where the two sides of the trapezoid are the buying range ($[z_1, z_2]$) and the selling range ($[z_3, z_4]$), respectively, and their slopes are used as the investment weights.

2.4 Fitness function

In order to find effective valuation rules under maximum return and minimum risk, the fitness function evaluates the return of the trading strategy induced by the Fuzzy expression tree. Given a fuzzy expression tree *E*, its fitness *f(E)* is derived from Equation (4), where *ROI* is return of investment from all trades and σ is standard deviation from net value of all of trade days. Let NV_i be the net value of *i*-th trades, $\overline{NV}$ is the mean of *NV* and N is the number of trades. σ is evaluated from Equation (5).

$$f(E) = \frac{ROI}{\sigma} \tag{4}$$

$$\sigma = \sqrt{\frac{\sum_{i-1}^{N} (NVi - \overline{NV})^2}{N}} \tag{5}$$

2.5 Fuzzy trading strategy

We proposed two fuzzy trading strategies to analysis ROI performances. They are Multi-Valued trading strategy based on trapezoidal fuzzy number and Singled-Valued trading strategy based on triangle fuzzy number.

- **Multi-Valued trading strategy**

The trapezoidal fuzzy number is used to stock valuation inducing a Multi-Valued trading strategy as shown in Figure 5. Each trapezoidal fuzzy number on Multi-Valued stock price is divided into three ranges: Buying Range, Selling Range and Intrinsic Value Range. Our trading strategies apply buying actions, selling actions and nothing to do on them, respectively. If the market price of stock (*p*) enters Buying Range, $x_1 \leqq p \leqq x_2$, the invested capital ratio is proportional to membership degree. On the contrary, if *p* enters Selling Range, $x_3 \leqq p \leqq x_4$, the sell shares ratio depends on membership degree. Finally, neither buying nor selling actions is used if p falls into Intrinsic Value Range, $x_2 \leqq p \leqq x_3$.

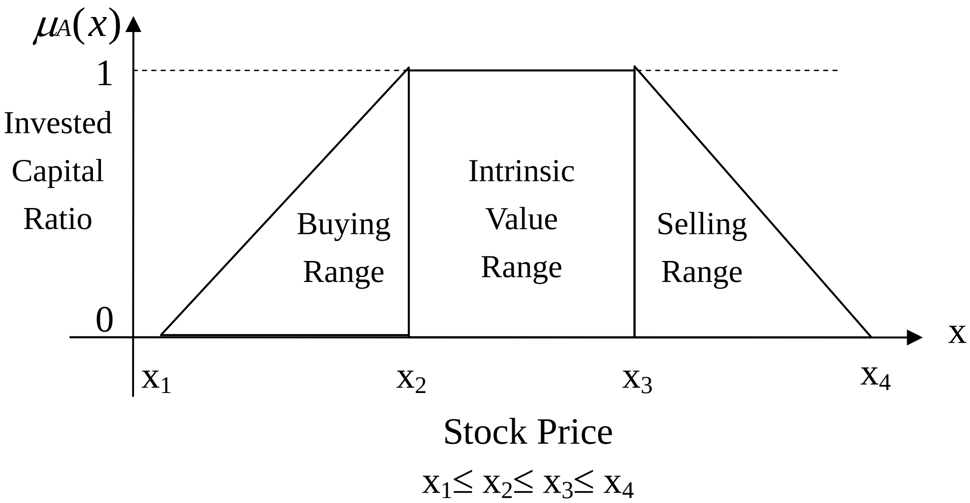


Figure 5: Multi-Valued trading strategy

- **Single-Valued trading strategy**

We also use triangle fuzzy number to stock valuation inducing a Single-Valued trading strategy as shown in Figure 6. The Single-Valued (triangle) strategy is a special case of Multi-Valued (trapezoidal) trading strategy illustrated in Figure 5. The difference of them is that the intrinsic value of stock is only a single value in this strategy. Each triangle fuzzy number is divided into two ranges: Buying Range and Selling Range. The similar buying actions and selling actions are used, if the market price of stock (p) enters Buying Range, $x_1 \leqq p \leqq x_2$, and Selling Range, $x_2 \leqq p \leqq x_3$, respectively.

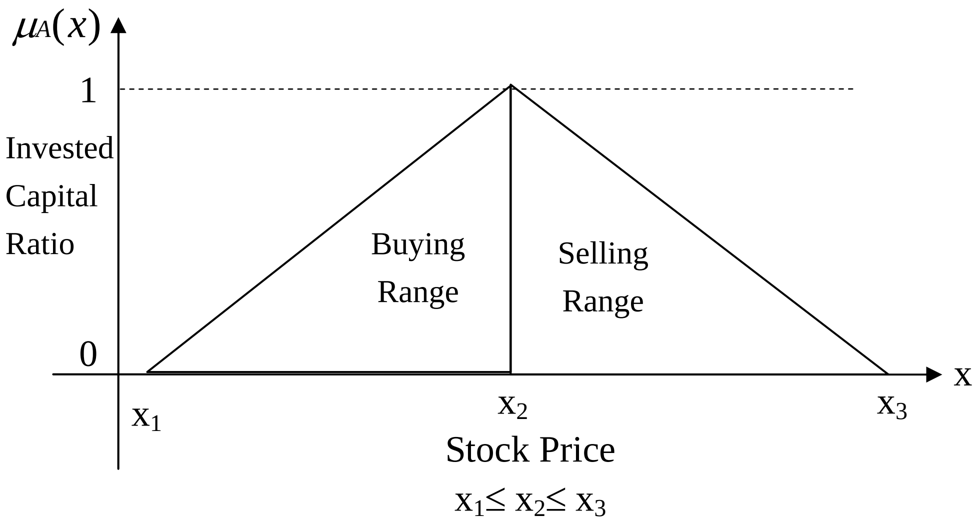


Figure 6: Single-Valued trading strategy

3 Experimental Results

The simulation environment, sample data and experimental results are described in this Section. Our FuzzyTree based program is written in Borland C++ Builder 6.0.

3.1 Fuzzy GP Parameters

The parameters used in our fuzzy GP runs shown in Table 2. The population size is 5000. The number of generations is set to 500. The selection method is tournament. The size of tournament is 2. The crossover method is our proposed FuzzyTree crossover. The crossover rate is set to 0.9 and mutation rate is set to 0.05. Minor changes in theses parameters seem not to have a major effect on the performance in our preliminary tries except crossover and selection.

Table 2: Parameters of Fuzzy GP

Number of generations	500
Population size	5000
Arithmetic operators	+, -, ×
Maximum tree depth	5
Crossover rate	0.9
Mutation rate	0.05
Selection	Tournament
Crossover	FuzzyTree Crossover
Mutation	Replacing a subtree
Reserved elitists	Three best individuals of each population

3.2 Sample Data

Eight electronic businesses are arbitrarily selected to be our testing targets as listed in Table 3. The mean and maximum total asset is $448.17 and $3005.28 hundred millions (United Micro Electronics). These companies have been listed and traded in the Taiwan Stock Exchange (TSE) since 1995 or earlier. The relevant data are collected from Taiwan Economic Journal Data Bank (TEJ).

Table 3: Descriptive statistics for our target companies Unit: 100 millions

Company	Market Value	Total Asset
Lite-On Electronics Co., Ltd. (LOE)	$869.86	$309.71
United Micro Electronics Co., Ltd.(UME)	4519.41	3005.28
Microtek Electronics Inc.	21.8	117.38

Delta Electronics, Inc.(Delta)	666.12	457.86
Advanced Semiconductor Engineering, Inc.(ASE)	948.77	669.25
Kinpo Electronics Inc.	250.21	199.89
Compeq Manufacturing Co., Ltd.(Compeq)	115.28	282.48
Hon Hai Precision Co., Ltd.	3552.04	1264.22
Mean	1439.09	448.17

The data encompasses the entire period from 1, January 1992 to 31, June 2003. The training phase (*k*), the test phase (*l*) and validation phase (*v*) are one period in each sliding window (*SW*) as show in Figure 7. *k*, *l* and *v* could be one month, one quarter (quarterly report), half a year or one year (annual report) according to financial statement report period. In this study, we use financial statement annual report to be our experiment data. The sliding window shifts one validation period gradually until the 31, June 2003. The $SW_{k\_l\_v}$ denotes the totally sliding window size: $k+l+v$. Among them, the training phase data is used to learn a stock valuation model; the test phase data is used to obtain multi-valued stock prices from the learned model; and the validation phase data is used for calculating ROI from according to trading strategies. It is noted that the financial statements announced to public generally delays half a year in Taiwan Stock Markets. Thus, the testing phase data in reality delays half a year to validation phase data in our experiments.

SW_1

Training	Test	Validation
k	*l*	*v*

SW_2

Training	Test	Validation
k	*l*	*v*

. . .

SW_i

Training	Test	Validation
k	*l*	*v*

Figure 7: Sliding windows simulation process

3.3 Analysis of results

For brevity, we summarize the performance of FuzzyTree crossover by three parts: (1) the comparisons of executing performances between FuzzyTree and subtree crossover, (2) the relationship between the size of sliding window and ROI and (3) the comparisons of ROIs between Multi-Valued and Single-Valued trading strategy. For avoiding outlier results in our experiments, each experiment is done five times and takes their mean value.

- **Executing Performance**

The objective of this study is to find a simple precise valuation model. So, the mean executing time and number of nodes are compared on FuzzyTree and subtree crossover methods, individually, which are shown in Table 4. The mean number of nodes used in FuzzyTree crossover (13.93) is less than that used in subtree crossover (23.08). In addition, the mean executing time of FuzzyTree crossover (00:04:07) is shorter than that of subtree crossover (00:16:54). It is obvious that our proposed FuzzyTree crossover method could find an effective and succinct valuation model quickly than subtree method.

Table 4: Executing performances (FuzzyTree vs. subtree crossover)

	FuzzyTree crossover		subtree crossover	
Company	No. of nodes	Exec. time	No. of nodes	Exec. time
Lite-On	15.8	03:51	18.4	15:22
UME	14.3	04:12	25.8	15:21
Microtek	16.2	04:23	23.3	15:34
Delta	14.6	04:22	27.9	15:19
ASE	13.7	04:08	19.3	15:33
Kinpo	11.1	03:53	22.2	15:11
Compeq	12.3	04:02	23.3	15:22
Hon Hai	13.4	04:05	24.4	15:24
Mean	13.93	04:07	23.08	16.54

- **Sliding Window**

Due to over-fit learning, the size of training phase (k) in sliding window would deeply influence final ROI. And, it also makes the change of sliding window size. In Table 5, we set test phase (l) = 1 and validation phase (v) = 1, and choose k=10, 3, 1, respectively, to compare their ROIs. The mean ROI of $SW_{10\_1\_1}$, $SW_{3\_1\_1}$, $SW_{1\_1\_1}$ are 4.39, 10.7, and 15.8 shown in Table 5. By implementing the sliding window size, we proved that shorter training phase explains more significant ROI. It appears that the long-term financial statement information could distort the intrinsic value of stock.

Table 5: Sliding windows size vs. ROI

Company	$SW_{10\_1\_1}$	$SW_{3\_1\_1}$	$SW_{1\_1\_1}$
Lite-On	7.25	4.01	10.92

UME	-10.04	-1.58	1.10
Microtek	0.62	76.92	35.65
Delta	24.29	20.41	34.49
ASE	-6.34	1.73	9.37
Kinpo	2.39	2.62	9.52
Compeq	1.15	-49.37	6.58
Hon Hai	15.83	30.88	18.79
Mean	4.39	10.7	15.8

- **Trading Strategy**

Two trading strategies: Multi-Valued and Single-Valued Strategies are proposed and integrated with our FuzzyTree crossover method to achieve better ROIs than using Buy-and-Hold strategy. The Buy-and-Hold strategy is a general comparison benchmark, which buys stocks at beginning, holds (nothing to do) until the ending of investment period, and sells them, regardless of rational stock prices.

Table 6 lists the obtained ROI using Buy-and-Hold trading strategy (*ROI-Buy-and-Hold*), Single-Valued trading strategy (*ROI-Single-Valued*) and Multi-Valued trading strategy (*ROI-Multi-Valued*), respectively. Because long-term statement would distort intrinsic value of stock, the selected training phase, test phase and validation phase are year 2000, 2001 and (2002.7.1 – 2003.6.30).

From Table 6, it is shown that the ROI of *Multi-Valued* strategy (16.20) is greater than using *Single-Valued* (4.26) and *Buy-and-Hold* (-36.89) strategies on all 8 companies. The similar results are also shown in Figure 8. It means that multi-valued price is more suitable for stock valuation and trading strategy than using single-valued price.

Table 6: ROI of Multi-Valued, Single-Valued and Buy-and-Hold strategies

Company	*ROI-Multi-Valued*	*ROI-Single-Valued*	*ROI-Buy-and-Hold*
Lite-On	22.75	8.59	-44.36
UME	2.05	-11.00	-53.83
Microtek	15.03	10.32	-13.51
Delta	27.56	4.49	-21.64
ASE	12.42	8.18	-35.27
Kinpo	13.25	2.09	-28.57
Compeq	17.26	7.26	-72.65
Hon Hai	19.25	4.18	-25.28
Mean	16.20	4.26	-36.89

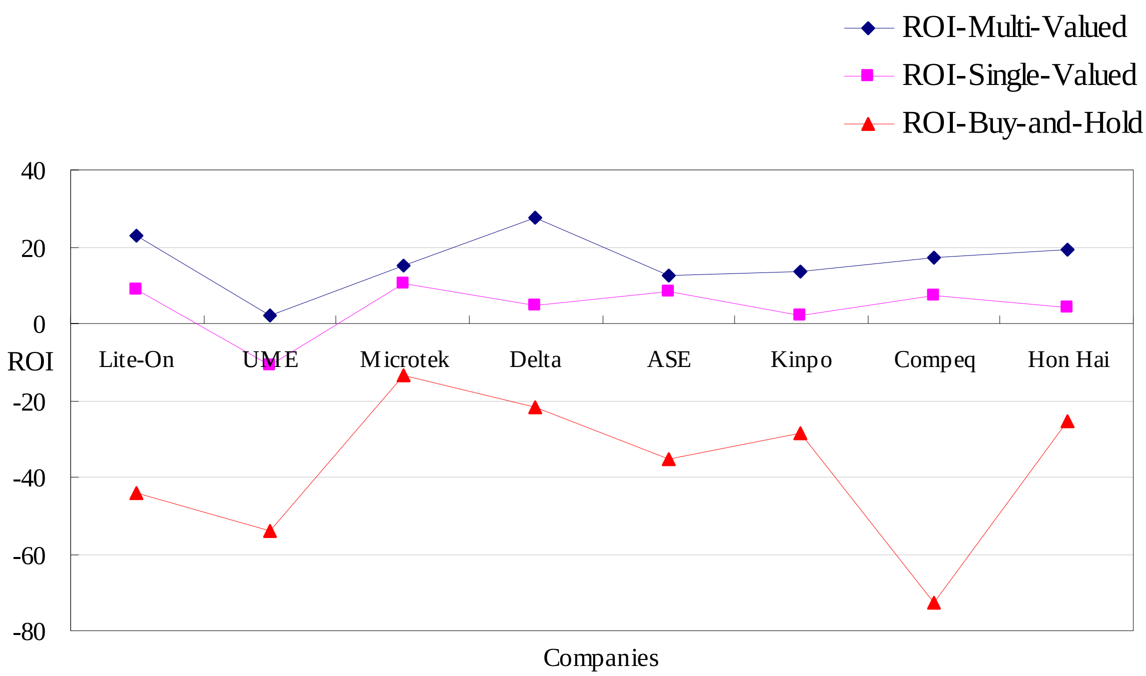


Figure 8: Comparison ROI performance of *Multi-Valued, Single-Valued and Buy-and-Hold* strategies

4 Conclusions

This paper describes a fuzzy GP multi-valued stock valuation model and a fuzzy tree crossover operator. Results on 8 arbitrarily selected electronic companies demonstrate the feasibility of multi-valued stock valuation model and the superiority of the FuzzyTree crossover operator. First, the number of tree nodes of FuzzyTree operator is simpler than those of subtree. Therefore, the run time of FuzzyTree is also shorter than that of subtree. The FuzzyTree crossover could found a succinct stock valuation model effective than subtree model. The FuzzyTree crossover seems to improve convergence during evolution. Second, from the results of different sliding window sizes, it seems that shorter training period produces better ROI. It indicates that long-term financial statement could distort the intrinsic value of stock. Finally, the ROI's of Multi-Valued strategy are greater than the ROI of Single-Valued and Buy-and-Hold. Clearly, this technique is a promising tool in multi-valued stock valuation.

References

[1]. L. Andersen. How Options Analysis Can Enhance Managerial Performance. European Management Journal, 20: 505-511, 2002.

[2]. B. S. Ahn, S. S. Cho, and C. Y. Kim. The integrated methodology of rough set theory and artificial neural network for business failure prediction. Expert Systems with Applications, 18:65-74, 2000.

[3]. B. Back, T. Laitinen, and K. Sere. Neural networks and genetic algorithms for bankruptcy predictions. Expert Systems with Applications, 11:407-413, 1996..

[4]. S. Bhattacharyya, O. V. Pictet, and G. Zumbach. Knowledge-intensive genetic discover in foreign exchange markets. IEEE Transactions on Evolutionary Computation, 6:169-181, 2002.

[5]. F. E. Block. A study of price to book relationship. Financial Analysts Journal, pages 63-73, 1995.

[6]. J. E. Boritz and D. B. Kennedy. Effectiveness of neural network types for prediction of business failure. Expert Systems with Applications, 9:503-512, 1995.

[7]. C. Carlsson, R. Fuller, and P. Majlender. A possibilistic approach to selecting portfolios with highest utility score. Fuzzy Sets and Systems, 131:13-21, 2002.

[8]. L. H. Chen and T. W. Chiou. A fuzzy credit-rating approach for commercial loans: A taiwan case. The International Journal of Management Science, 27:407-419, 1999.

[9]. S. H. Chen, W. C. Lee, and C. H. Yeh. Hedging derivative securities with genetic programming. International Journal of Intelligent Systems in Accounting, Finance and Management, 8:237-251, 1999.

[10]. S. H. Chen and C. H. Yeh, Toward a Computable Approach to The Efficient Market Hypothesis: An Application of Genetic Programming", Journal of Economic Dynamics and Control, 21:1043-1063, 1997.

[11]. J. R. Dietrich, M. S. Harris and K. A. Muller III. The reliability of investment property fair value estimates. Journal of Accounting and Economics, 30:125-158, 2001.

[12]. E. F. Fama and K. R. French. Size and book-to-market factors in earnings and returns. Journal of Finance, 50:131-155, 1995.

[13]. J. Francis, P. Olsson, and D. R. Oswald. Comparing the accuracy and explainability of dividend, free cash flow, and abnormal earnings equity value estimates. Journal of Accounting Research, 38 (Spring):45-70, 2000.

[14]. M. I. Heywood and A. N. Zincir-Heywood. Dynamic Page Based Crossover in Linear Genetic Programming, IEEE Transaction on Systems, Man, and Cybernetics-Part B: Cybernetics, 32, pages 380-388, 2002.

[15]. M. J. Heywood and A. N. Zincir-Heywood. Page-Based Linear Genetic Programming, IEEE International Conference on Systems, Man, and Cybernetics, pages 3823-3828, 2000.

[16]. J. Koza., Genetic Programming, Cambridge, Mass.: MIT Press 1995.

[17]. I. Makoto, and I. Hitoshi. Island Model GP with Immigrants Aging and Depth-Dependent Crossover. In Proceeding of IEEE 2002 International Conference on Evolutionary Computation, pages 267-272, 2002.

[18]. T. H. Payne and J. H. Finch. Effective teaching and use of the constant growth dividend discount model. Financial Services Review, 8:283–291, 1999.

[19]. H. Tanaka, H. Nakayama, and A. Yanagimoto. Possibility portfolio selection. In Proceedings of 1995 IEEE International Joint Conference on Fuzzy Systems, pages 813-818, 1995.

[20]. ITO. Takuya, IBA Hitoshi and SATO Satoshi, Depth-Dependent Crossover for Genetic Programming, in Proceedings of IEEE 1998 International Conference on Evolutionary Computation, pages 775-780, 1998.

[21]. M. D. Terrio and M. I. Heywood, Directing Crossover for Reduction of Bloat in GP. In Proceedings of the IEEE 2002 Canadian conference of Electrical & Computer Engineering, pages 1111-1115, 2002.

[22]. O. Una-May and O. Franz, Hybridized Crossover-Based Search Techniques for Program Discovery, IEEE 1995 International Conference on Evolutionary Computation, pages 575-582, 1995.