Adaptive Genetic Programming for Option Pricing

Zheng Yin
Natural Computing Research
and Applications Group
University College Dublin
Dublin, Ireland
zheng.yin@ucdconnect.ie

Anthony Brabazon
Natural Computing Research
and Applications Group
University College Dublin
Dublin, Ireland
anthony.brabazon@ucd.ie

Conall O'Sullivan
School of Business
University College Dublin
Dublin, Ireland
conall.osullivan@ucd.ie

ABSTRACT

Genetic Programming (GP) is an automated computational programming methodology, inspired by the workings of natural evolution techniques. It has been applied to solve complex problems in multiple domains including finance. This paper illustrates the application of an adaptive form of GP, where the probability of crossover and mutation is adapted dynamically during the GP run, to the important real-world problem of options pricing. The tests are carried out using market option price data and the results illustrate that the new method yields better results than are obtained from GP with fixed crossover and mutation rates. The developed method has potential for implementation across a range of dynamic problem environments.

Categories and Subject Descriptors

J.m [**Miscellaneous**]: Finance; I.2.6 [**Artificial Intelligence**]: Parameter learning

General Terms

Economics

Keywords

Genetic Programming, option pricing

1. INTRODUCTION

Recent years have seen the application of multiple biologically-inspired algorithms for the purposes of financial modeling [1]. A number of these algorithms draw metaphorical inspiration from processes of natural evolution. One of the most studied evolutionary methodologies is that of genetic programming (GP) [12]. GP is a population-based search algorithm. It starts from a high-level statement of what is required and automatically creates a computer programme to solve the problem. GP belongs to the field of *Evolutionary Automatic Programming.* The term is used to refer to systems that adopt evolutionary computation to automatically generate computer programmes. More generally, a computer programme can be considered as a list of rules or as a model. In this study, we initially review the application of genetic programming (GP) for options pricing and propose a new adaptive method to improve the results of GP for this problem.

1.1 Exchange Traded Options

Exchange traded options have been actively traded on stocks, stock indices, foreign currencies and futures contracts by hedgers, speculators and arbitrageurs since the 1970s [4]. An option can be defined as the right but not the obligation to buy or sell a financial asset at a stated price at or during a specified time window. Option prices are affected by multiple factors. The first and most well known option pricing model is the Black-Scholes model. This model assumes the underlying asset returns follow an arithmetic Brownian motion which results in normally distributed returns at every horizon. The Black-Scholes model is a simplified version of reality and when used to fit market options data it performs relatively poorly because of its underlying assumptions. These assumptions include: that asset prices follow geometric Brownian motion; that asset prices cannot experience discontinuous jumps; that the volatility of asset prices is constant; and that one can trade the financial asset continuously and therefore that options can be perfectly hedged by trading in the underlying asset and a risk-free bond.

The flaws in the Black-Scholes model have encouraged the development of multiple new approaches to options pricing. Due to the complexity in developing closed form theoretical models for options pricing, the domain is particularly amenable to techniques such as GP.

One of the early applications of GP to the option pricing problem is provided by [19]. Since then there have been many improvements such as seeding the initial population with elements drawn from the Black-Scholes option pricing formula, and the combination of other domain knowledge into the GP set of terminals / functions[21]. In this paper a new adaptive GP method is proposed where the probability of crossover and mutation is dynamically altered during the GP run. The method is benchmarked against a fixed parameter GP system using market option prices.

1.2 Motivation

The problem of options pricing is a particularly suitable domain for the application of GP. GP is especially useful in data rich environments; where the search space is large and highly complex; where conventional mathematical analysis

GECCO'07, July 7–11, 2007, London, England, United Kingdom.
Copyright 2007 ACM 978-1-59593-698-1/07/0007 ...$5.00.

cannot provide analytic solutions; and where the interrelationships among the relevant variables are poorly understood. In options pricing there are multiple factors that effect the option price thus the search space is large. The interrelationships of these factors are also complex and nonlinear. Current option pricing theory has obvious biases that usually lead to model option prices that can differ systematically from market option prices. Certain options such as exotic option have no exact pricing formula thus GP has obvious utility in these situations. Compared with recent applications of GP to options pricing, this paper focuses on how to apply GP in a more efficient way not only by embedding domain knowledge of option price characteristics in the GP but also by allowing parameter settings to adapt during the GP run.

1.3 Structure of Paper

The first two sections of this paper provide an introduction to options pricing and describe previous work in applying GP to options pricing. The adaptive GP methodology used in this paper is then introduced. Next, the experimental design and key results are outlined. The final section of the paper contains conclusions and suggestions for future work.

2. OPTIONS PRICING

An option contract gives the holder the right to buy or sell a certain amount of the underlying asset by a certain date (maturity date) for a certain price known as the strike price. The option price is sensitive to a number of factors. The non-linear relation between the option price and these factors further contributes to the complexity of options pricing. For example, in the Black-Scholes setting for a non-dividend paying European call option there are five factors that affect the price of the option. These are the:

1. current stock price S_0
2. strike price K
3. volatility of the stock return, σ
4. risk-free rate r
5. time to maturity T

As briefly mentioned in section 1, the Black-Scholes model embeds several key assumptions. The stock price undergoes a diffusion process thus is log normally distributed with an instantaneous drift and volatility given by μ and σ respectively. The volatility σ and the risk-free rate r are constant during option life. This implies:

$$\ln \frac{s_T}{s_0} \sim n[(\mu - \frac{\sigma^2}{2})T, \sigma\sqrt{T}] \tag{1}$$

This assumption leads to the following:

$$S_T = S_0 e^{\eta\ T}, \eta \sim n[(\mu - \frac{\sigma^2}{2}), \frac{\sigma}{\sqrt{T}}] \tag{2}$$

μ is the instantaneous expected return on the stock
σ is the instantaneous volatility of stock price return
S_T is the stock price at a future time T
S_0 is the stock price at time zero
$n(m, s)$ denotes the normal density function with mean m and standard deviation s
η is defined as the continuously compounded rate of return per annum realized between time zero and T

The Black-Scholes formula for the price at time zero of an European call option on a non-dividend paying stock is:

$$c = S_0 N(d_1) - Ke^{-rT} N(d_2) \tag{3}$$

$$d_1 = \frac{\ln(S_0/K) + (r + \sigma^2/2)T}{\sigma\sqrt{T}} \tag{4}$$

$$d_2 = d_1 - \sigma\sqrt{T} \tag{5}$$

$N(\chi)$is the cumulative normal probability distribution function for a standardized normal distribution
K is the strike price
r is risk-free interest rate
c is the European call option price

The key assumption in the Black-Scholes model is that the underlying stock price is log normally distributed and the volatility is constant. These assumptions do not hold true in reality as the distribution of price changes has fatter tails than those implied by a log normal distribution, and the volatility changes over time. Thus serious biases show up in the Black-Scholes model when it is used to price options that trade in markets. This is reflected when one calculates the volatility that is substituted into the Black-Scholes pricing formula so that the BS option price is equal to the market price. This is called the Black-Scholes implied volatility. If the assumptions underlying the BS option pricing model were correct, the BS implied volatilities for options on the same underlying asset would be constant for different strike prices and maturities. However in practice the BS implied volatilities are varying over strike price and maturity. Given that the options are written on a single underlying asset this result seems at first paradoxical, i.e. we have a number of different implied volatilities for a single asset which should only have one measure for its volatility. Yet if we relax some of the assumptions in the BS model, such as allowing for a more complex data generating process for the asset price than the log normal stochastic process, and if we take into account the resulting complications, this result begins to make sense. It highlights the erroneous assumptions that underpin the BS model.

3. GP AND OPTIONS PRICING

In applications of GP to options pricing, the objective is to uncover the Black-Scholes option pricing formula, using market data. The utility of the formula is typically tested by comparing the quality of its predictions against real market option prices.

In this study, data is drawn from market option prices on the FTSE 100 on the 17th March 2006. The options used are call and put options on the FTSE 100 futures index. The data supplied was in the form of implied volatilities. There 187 different end-of-day settlement implied volatilities quoted for various strike prices and maturities. The option *moneyness* (defined as the the underlying asset price divided by the strike price) in our 187 data points varies from 0.77 to 1.4 and the time-to-maturity varies from 35 days to 5754 days. Market call option prices are calculated by substituting the implied volatilities into the Black-Scholes formula. The object of the GP application therefore, is to try and recover the Black-Scholes formula for call options that will

Table 1: Terminal set

$ExpressionSign$	$Expression$	$Definiton$
X1	S_0/K	Asset price / Strike price
X2	$r * T$	time to maturity times risk free rate
X3	$\sigma * \sqrt{T}$	implied asset price volatility during option's life

Table 2: Non-terminal set

$Expression\ Sign$	$Definiton$
+	Addition
-	Subtraction
*	Multiplication
x/y	Protected division, if y=0 then x/y= x, if y $\neq$ 0 then x/y=x/y
x^2	Square
$N(x)$	Accumulated normal distribution
e^x	Exponential Function

reproduce the entire set of the 187 option prices given the explanatory variables.

3.1 Terminal and Function Set Selection

In selecting terminals and functions for the experiments, explanatory variables that might affect the option prices should be included in the terminal set. The function set should include all the potential mathematical or logic functions that could be used in options pricing formula. In selecting variables for inclusion as terminals, we used domain knowledge [21] to include option moneyness, the risk-free rate scaled by maturity (defined as the time to maturity multiplied by the risk free rate of return) and implied volatility during the life of the option (see table 1). The non-terminal set is as listed in table 2.

3.2 Fitness Function

The evolved models are designed to predict market option prices across a range of strike prices and time-to-maturity periods. The most common fitness functions for options pricing are the mean squared, average absolute, and percentage error. It should be noted that when the option is deep in the money the absolute error is large compared with the percentage error, but when the option is out of money the option value is very small and near zero. The absolute error is typically much smaller compared with the percentage error. The fitness function in eq. 6 ensures that the model works well for both in the money and out of money options.

$$fitness = \sqrt{\frac{\sum(R_{GP} - P_M)^2 + \sum(\frac{R_{GP}-P_M}{P_M})^2}{2 \times N}} \tag{6}$$

where R_{GP} is the result evaluated from the function tree (each individual in a population) created in GP, P_M is the market option price divided by its corresponding strike price, and N is the number of the data observations which in this case is 187.

3.3 Error Measurement

Our target set here is the Black-Scholes call option price divided by the strike price. Ultimately with a view to trading options, it is hard to directly infer the financial utility of a GP evolved formula with the fitness metric used in the GP application. To make the results more intuitive we calculate and report two error measures for the GP runs, namely average absolute error and percentage absolute error.

1. **Absolute Error Measurement**

$$A.E. = \frac{\sum \sqrt{(P_{GP} - P_M)^2}}{N} \tag{7}$$

2. **Percentage Error Measurement**

$$P.E. = \frac{\sum \frac{\sqrt{(P_{GP}-P_M)^2}}{P_M}}{N} \tag{8}$$

where $P_{GP} = R_{GP} * K$ i.e. P_{GP} is the GP option price and P_M is the market option price.

4. PARAMETER VALUES FOR CROSSOVER AND MUTATION

A practical issue when applying GP to real-world problems is making good parameter choices for the GP. Many options are open to the modeller including the:

- form and rate of mutation
- form and rate of crossover
- form and pressure of selection mechanism
- form of replacement mechanism
- size of population

Even when attention is restricted to the choice of good crossover and mutation rates, the modeller faces a non-trivial problem. A common approach in tuning GP is to undertake a series of trial and error experiments, before making parameter choices for the final GP runs. However, this approach is highly problematic as it can be time consuming and it is impractical to test all parameter choices. Another issue is that good choices for parameters such as crossover and mutation rates are unlikely to remain constant over the entire run. Rather than selecting static parameter values,

Table 3: The error results from the constant parameter GP

Run	1	2	3	4	5	6	7	8	9	10	Average
P.E. (%)	18.7	22	144	28.3	35.1	48.5	32.7	29.1	48.2	32.3	43.9
A.E.	87.8	99	151	170	178	178	186	189	197	204	163.9

Table 4: The error results from the adaptive parameter GP

Run	11	12	13	14	15	16	17	18	19	20	Average
P.E. (%)	14.2	16.9	21.1	17.1	22.3	42.9	34.7	24.4	28.1	31.8	25.3
A.E.	42.9	55.5	84.7	95.1	99	133	137	147	157	189	114

another approach is to co-evolve the parameters during the run. Three broad methods of such adaptation exist (see Fig. 1) [3]. In this study we adopt the second approach.

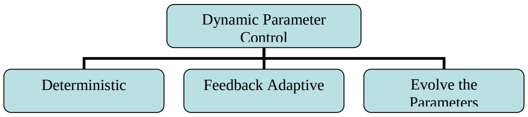


Figure 1: Taxonomy of adaptive parameter control

Deterministic methods of parameter control vary parameter settings during the GP run, without using any feedback from the search process. A simple example of a deterministic rule for a mutation value would be:

$$\alpha(t) = \alpha(t^0)(1 - 0.8\frac{t}{T}) \quad (9)$$

where t $(0 \leq t \leq T)$ denotes the current generation and $\alpha(t^0)$ is a fixed value. This rule will reduce the value of the mutation rate during the run.

Under a feedback adaption process, the parameter values are adapted based on feedback from the algorithm. For example, if the intent was to increase the level of diversity generation once the population has converged to a threshold level (perhaps measured using the entropy of the population of binary strings), once such convergence is detected, the mutation rate could be increased by x%. Another possibility is to evolve good choices for its parameters dynamically during the run.

The parameter choices for crossover and mutation are clearly critical in ensuring a successful GP application. It impacts on populational diversity and the ability of GP to escape from local optima. Optimal parameter setting is also linked to the complexity of the problem and to the size of the population. If the search space is large and/or the population size is relatively small, then the mutation rate will need to increase. In previous applications of GP to options pricing, the probabilities of crossover and mutation were typically kept constant. For example, in [19] mutation is applied at a rate of 0.0033 with a level of 0.001 being applied by [15] and [20]. Chidambaran [14] investigated the utility of various mutation rates between 0.1 to 0.5 (each of which was constant in a single run).

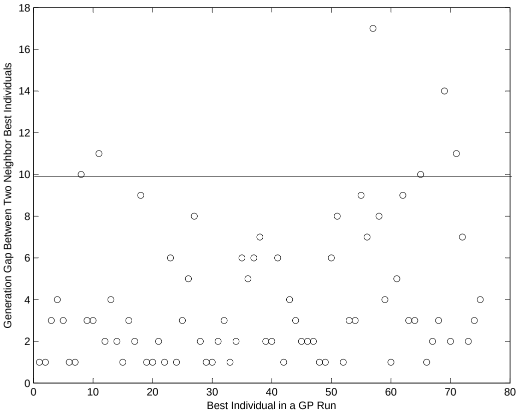
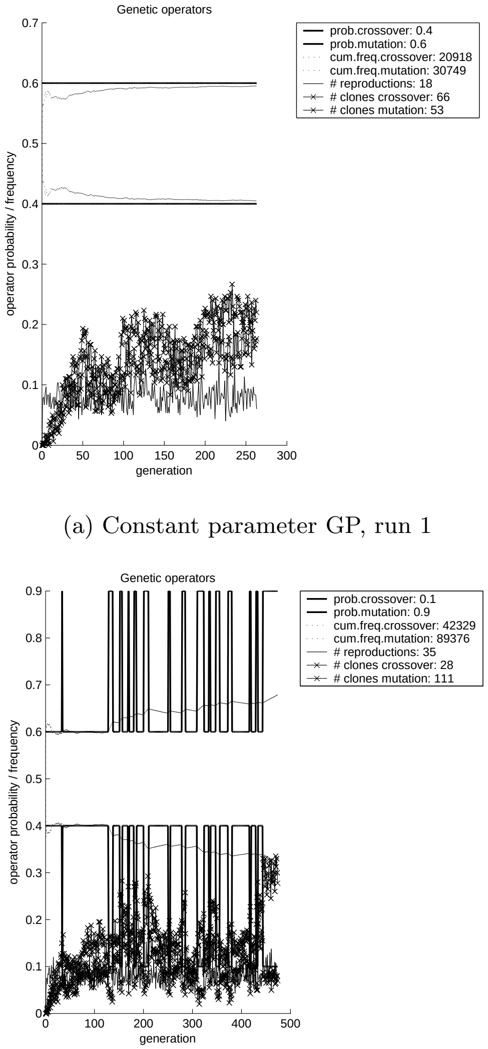


Figure 2: Generation Gaps Between Successive Best Individuals

As already noted, in this study we employ three explanatory variables, stock price divided by strike price, risk free rate times options' time to maturity, underlying variance, and there are seven functions in the non-terminal set. Based on initial experiments, it was found that the best individual changes frequently in early generations, with the rate of change slowing down later in the run. Generation gaps between new best individuals are plotted in figure 4. The points above the line in the graph indicate the generation gap between subsequent best individuals are more than 10 generations. Based on these results we set a window size of 10 generations. In future applications, window size could be determined dynamically during the GP run.

In adapting the mutation parameter, we steadily increase its value whenever the best individual is unchanged over several generations. The parameter then reverts to its initial level a new best individual is uncovered by GP. This allows the search process to adapt to escape local optima, whilst permitting local improvement around just discovered new solutions. In the experiments, the width of the window is set at ten generations. In other words, the mutation parameter is fixed for the ten generations following the uncovering of a new best solution. After ten generations without finding a new best individual, the mutation probability increased until



(a) Constant parameter GP, run 1

(b) Adaptive parameter GP, run 11

Figure 3: Crossover and Mutation parameters from run 1 and 11

either it reaches 0.9 or alternatively, a new best solution is uncovered. Figures 3(a) and 3(b) illustrate the adaption of crossover and mutation rates during a sample GP run.

5. EXPERIMENTAL DESIGN

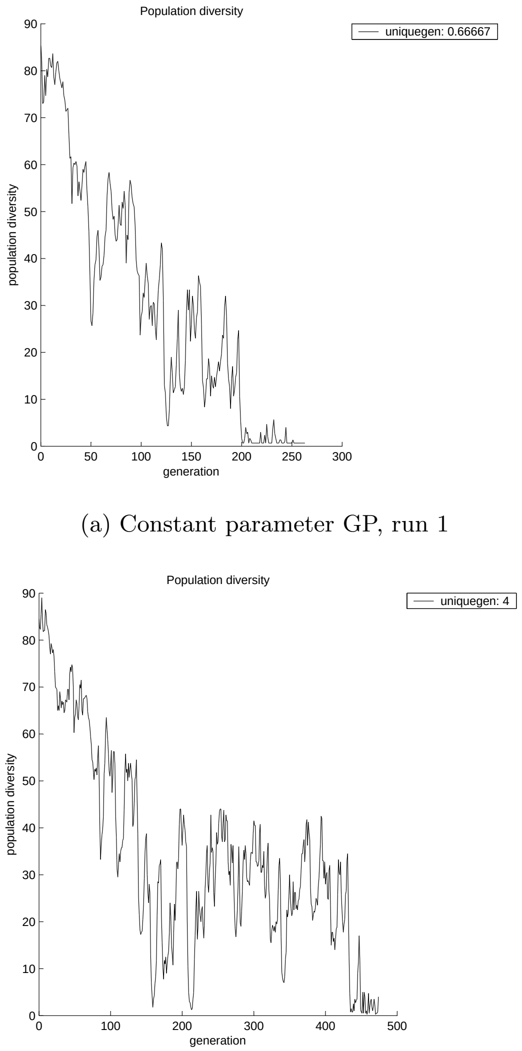
A total of twenty GP runs were undertaken, ten of which were fixed parameter GPs and ten were dynamic parameter GPs. The fixed parameters for crossover and mutation were 0.4 and 0.6 respectively, set after some initial trial and error experiments. In the adaptive experiments, the parameters for crossover and mutation are initially set to 0.4 and 0.6. If ten generations have elapsed and the best individual has not changed this means the population is perhaps too concentrated, hence the mutation rate is ncreased by 0.02 per generation, with crossover decreasing by 0.02 each time until limits of 0.9 and 0.1 are reached. Once a new best individual appears the mutation and crossover rates are put back to their initial values of 0.6 and 0.4.

For all experimental runs, ramped half-half initialization is employed. A roulette parental selection with a replacement strategy of half elitism, which means half of the new population will be filled by the best from both parent and children and the remaining places will be left to the best children, is also employed. The population size is fixed at 300. The GP run is terminated either when there has been no performance improvement for 40 generations, with the maximum number set to 800 generations.

6. RESULTS

The results from the twenty GP runs are provided in tables 3 and 4. The average absolute error from the constant parameter GP is 44% higher than the dynamic parameter GP counterpart, and the average percentage error is 73% higher. These results suggest that significant improvements are made with the use of a dynamic parameter GP.

It can be seen from the population diversity graphs, figures 4(a) and 4(a), generated in the GP run, that the population in the dynamic parameter GP is quiet diversified in the middle of the run compared to the constant parameter GP where the population setting decreases monotonically.



(a) Constant parameter GP, run 1

(b) Adaptive parameter GP, run 11

Figure 4: Population Diversity from run 1 and 11

7. CONCLUSIONS

This paper illustrates the application of an novel adaptive form of GP, where the probability of crossover and mutation is adapted dynamically during the GP run, to the important real-world problem of options pricing. The tests are carried out using market option price data and the results illustrate that the new method yields better results than are obtained from GP with fixed crossover and mutation rates.

Future work includes correcting the biases in the Black-Scholes options pricing model by applying GP to recover market option prices across a range of strike prices and maturities without using every implied volatilitiy in the sample and thereby examining the interrelationships among option prices that goes beyond the Black-Scholes pricing formula. It is also noted that the adaptive GP method has potential for implementation across a wide range of dynamic problem environments and it is intended to test the utility of the methodology on a variety of non-financial dynamic problems.

8. REFERENCES

[1] Brabazon A.,O'Neill M. (2006). *Biologically Inspired Algorithms for Financial Modelling*, Springer, Berlin.

[2] Brabazon A.,O'Neill M. (2004). Evolving Technical Trading Rules for Foreign-Exchange Markets, *Computational Management Science*,Vol.1, No.3-4, pp. 311-327.

[3] Eiben A.E.,Smith J.E. (2003). *Introduction to evolutionary computing*, Springer, Berlin.

[4] Hull J.C. (2003). *Options, Futures, and other Derivatives*, Pearson Education, New Jersey.

[5] Keber C.,Schuster M.G. (2002). Option Valuation with Generalized Ant Programming, *Genetic and Evolutionary Computation Conference 2002*, pp.74-81.

[6] Langdon W.B.,Poli R. (2002). *Foundations of Genetic Programming*, Springer-Verlag, Berlin.

[7] Luke S.,Panait L. (2002). Lexicographic Parsimony Pressure, *Genetic and Evolutionary Computation Conference, 2002*, Morgan Kaufmann Publishers, San Francisco, CA., pp. 829-836.

[8] Holland J.H. (1975). *Adaption in Natural and Artificial Systems*, University of Michigan Press.

[9] Chen S.H. (1998). *Evolutionary Computation in Financial Engineering*: A Roadmap of GAs and GP, Financial Engineering News, No.6, Sep. 1998.

[10] Chidambaran N.K.,Lee C.J. and Trigueros J.R. (1998). An Adaptive Evolutionary Approach to Option Pricing via Genetic Programming, New York University, Leonard N. Stern School Finance Department Working Paper Series, New York University, Leonard N.Stern School of Business, 98-086.

[11] Koza, J.R. (1994). *Genetic Programming II: Automatic Discovery of Reusable Programs.* MIT Press.

[12] Koza, J.R. (1992). *Genetic programming - on the programming of computers by means of natural selection*, MIT Press.

[13] Chidambaran N.K.,Triqueros J.R.,Lee C.W.J. (2002). Option Pricing Via Genetic Programming in *Evolutionary Computation In Economics And Finance*,2002, Physica-Verlag Heidelberg, pp.383-397.

[14] Chidambaran N.K. (2003). Genetic Programming with Monte Carlo Simulation for Option Pricing,*Proceedings of the 2003 Winter Simulation Conference*. Chick S.,Sanchez P.J., Ferrin D. and Morrice D.J.,eds, Vol.1, pp. 285-292

[15] Noe T.H.,Wang J. (2002). The Self-Evolving Logic Of Financial Claim Prices in *Genetic algorithms and genetic programming in computational finance*, Kluwer Academic Publishers, pp. 249-279.

[16] Chen S.H., Kuo T.W. and Shieh Y.P. (2002). Genetic Programming: A Tutorial with the Software Simple GP in *Genetic algorithms and genetic programming in computational finance*, Kluwer Academic Publishers,Dordrecht,2002, pp.55-77.

[17] Chidambaran N.K.,Triqueros J.R. and Lee C.-W.J. (2002). An Adaptive Evolutionary Approach to Option Pricing via Genetic Programming, in: Koza J., *Genetic Programming 1998: Proceedings of the Third Annual Conference*, Morgan Kaufmann, San Francisco, CA, pp. 187-192.

[18] Triqueros J.R. (1997). A Nonparametric Approach to Pricing and Hedging Derivative Securities via Genetic Regression, in *Proceedings of the IEEE/IAFE 1997 Computational Intelligence for Financial Engineering*, IEEE Press, pp. 1-6.

[19] Chen S.H.,Yeh C.H. and Lee W.C. (1998). Option Pricing with Genetic Programming, in *proceedings of the Third Annual Genetic Programming Conference*, Morgan Kaufmann Publishers, San Francisco, CA, pp. 32-37.

[20] Keber C.(2002). Evolutionary Computation in Option Pricing: Determining Implied Volatilities Based on American Put Option. in *Evolutionary Computation In Economics And Finance*,Physica-Verlag, pp. 399-415.

[21] Yin Z.,Brabazon A.,O'Sulivan C. (2006). Genetic Programming and and Option Pricing. *Proceedings of 2006 Annual Irish Accounting & Financial Association Conference.*