Quantum-Inspired Evolutionary Algorithms for Calibration of the VG Option Pricing Model

Kai Fan[1], Anthony Brabazon[1], Conall O'Sullivan[2], and Michael O'Neill[1]

[1] Natural Computing Research and Applications Group
University College Dublin, Ireland
kai.fan@ucd.ie, Anthony.Brabazon@ucd.ie, Michael.ONeill@ul.ie
[2] School of Business, University College Dublin, Ireland
conall.osullivan@ucd.ie

Abstract. Quantum effects are a natural phenomenon and just like evolution, or immune systems, can serve as an inspiration for the design of computing algorithms. This study illustrates how a quantum-inspired evolutionary algorithm can be constructed and examines the utility of the resulting algorithm on Option Pricing model calibration. The results from the algorithm are shown to be robust and comparable to those of other algorithms.

1 Introduction

The objective of this study is to illustrate the potential for using a quantum rather than a traditional encoding representation in an evolutionary algorithm, and also to assess the utility of the resulting algorithm for the purposes of calibrating an option pricing model. This purpose of this paper is to test the QIEA on a relatively simple option pricing model with the intention of testing the algorithm on more comprehensive option pricing models using more option data at a later stage.

In recent years there has been a substantial interest in the theory and design of quantum computers, and the design of programs which could run on such computers. One interesting strand of research has been the use of natural computing (for example GP) to generate quantum circuits or programs (algorithms) for quantum computers [1]. There has also been associated work in a reverse direction which draws inspiration from concepts in quantum mechanics in order to design novel natural computing algorithms. This is currently an area of active research interest. For example, quantum-inspired concepts have been applied to the domains of evolutionary algorithms [2,3,4,5,6], social computing [8], neurocomputing [9,10,11], and immuno-computing [12,13]. A claimed benefit of these algorithms is that because they use a quantum representation, they can maintain a good balance between exploration and exploitation. It is also suggested that they offer computational efficiencies as use of a quantum representation can allow the use of smaller population sizes than typical evolutionary algorithms.

Quantum-inspired algorithms offer interesting potential. As yet, due to their novelty, only a small number of recent papers have implemented a QEA, typically

M. Giacobini et al. (Eds.): EvoWorkshops 2007, LNCS 4448, pp. 189–198, 2007.

reporting good results [5,6]. Consequently, we have a limited understanding of the performance of these algorithms and further testing is required in order to determine both their effectiveness and their efficiency. It is also noted that although a wide-variety of biologically-inspired algorithms have been applied for financial modelling [7], the QEA methodology has not yet been applied to the finance domain. This study addresses both of these gaps.

2 The Quantum-Inspired Genetic Algorithm

The best-known application of quantum-inspired concepts in evolutionary computing is the quantum-inspired genetic algorithm (QIGA) [2,5,6]. The (QIGA) is based on the concepts of a qubit (quantum bit) and the superposition of states. In essence, in QIGAs the traditional representations used in evolutionary algorithms (binary, numeric and symbolic) are extended to include a quantum representation. Under a quantum representation, the basic unit of information is no longer a bit which can assume two distinct states (0 or 1), but is a quantum system. Hence, a qubit (the smallest unit of information in a two-state quantum system) can assume either of the two ground states (0 or 1) or any superposition of the two ground states (the quantum superposition). A qubit can therefore be represented as

$$|q^i\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

where $|0\rangle$ and $|\rangle$ are the ground states 0 and 1, and α & β are complex numbers that specify the probability amplitudes of the two ground states. The act of observing (or measuring) a qubit projects the quantum system onto one of the ground states. $|\alpha|^2$ is the probability that the qubit will be in state 0 when it is observed, and $|\beta|^2$ is the probability that it will be in state 1. Hence, a qubit encodes the *probability* that a specific ground state will be seen when an observation takes place, rather than encoding the ground states themselves. In order to ensure this probabilistic interpretation remains valid, the values for α and β are constrained such that $|\alpha|^2 + |\beta|^2 = 1$.

More generally, a quantum system of m qubits can represent a total of 2^m states simultaneously. In the language of evolutionary computation a system of m qubits can be referred to as a *quantum chromosome* and can be written as a matrix

$$\begin{bmatrix} \alpha_1 \; \alpha_2 \; \ldots \; \alpha_m \\ \beta_1 \; \beta_2 \; \ldots \; \beta_m \end{bmatrix} \tag{2}$$

A key point when considering quantum systems is that they can compactly convey information on a large number of possible system states. In classical bit strings, a string of length n can represent 2^n possible states. However, a quantum space of n qubits has 2^n *dimensions*. This means that even a short qubit can convey information on many possible system states. For example, a 3 bit quantum system can encode 8 (2^3) distinct binary strings, and an 8 bit quantum system can encode 256 distinct strings. Due to its probabilistic interpretation, a single qubit of length m can simultaneously represent *all* possible bit strings of length

2^m. This implies that it is possible to modify standard evolutionary algorithms to work with a single quantum individual, rather than having to use a population of solution encodings. The qubit representation of the system states can also help maintain diversity during the search process of an evolutionary algorithm, due to its capability to represent multiple system states simultaneously.

2.1 The Algorithm

There is no single QIGA, rather there are a family of possible algorithms which could be derived from the joint quantum-evolutionary metaphor. However, the following algorithm provides an example of a canonical QIGA

```
Set t=0
Initalise Q(t)

Create P(t) by undertaking an observation of Q(t)

Evaluate P(t) and select the best solution

Store the best solution in P(t) into B(t)

While (t < max t)
    t=t+1
    Create P*(t) by undertaking observations of Q(t-1)
    Evaluate P*(t)
    Update Q(t)
    Store the best solutions in B(t-1) and P(t) into B(t)
Endwhile
```

Initially, the population of quantum chromosomes is created

$$Q(t) = q_1(t), q_2(t), \ldots, q_n(t),$$

where n is the population size, and each member of the population consists of an individual qubit of length m. The α and β values for each qubit are set to $\frac{1}{\sqrt{2}}$ in order to ensure that the states 0 and 1 are equally likely for each qubit.If there is domain knowledge that some states are likely to lead to better results, this can be used to seed the initial quantum chromosome(s). Once a population of quantum chromosomes are created, these can be used to create a population of binary (or solution encoding) strings by performing an 'observation' on the quantum chromosomes. One way of performing the observation step is to draw a random number $rnd \in [0, 1]$. If $rnd > |\alpha_i(t)|^2$, the corresponding bit (j) in $p_i^j(t)$ is assigned state 1, otherwise it is assigned state 0. Due to the stochastic nature of the observation step, the QIGA could be implemented using a single quantum chromosome, where this chromosome is observed multiple times in order to generate the population $P(t) = p_1(t), p_2(t), \ldots, p_i(t)$. Alternatively, a small population of quantum chromosomes could be maintained, with each chromosome being observed a fixed number of times in order to generate $P(t)$. In the while loop, an update step is performed on the quantum chromosome(s). This update step could be performed in a variety number of ways, for example by using pseudo-genetic operators, or by using a suitable quantum gate [3]. However the step is undertaken, its essence is that the quantum chromosome is adjusted in order to make the generation of the best solution found so far, more likely in the next iteration. As the optimal solution is approached by the QIGA system, the values of each element of the quantum chromosome tend towards either 0 or 1, corresponding to a high probability that the quantum chromosome will generate a specific solution vector (p_i) when observed.

Quantum Mutation. Quantum mutation is loosely inspired by the standard GA mutation operator. However, this is adapted so that the mutation step is guided by the best individual found to date, with the quantum chromosome being altered in order to make the generation of this solution more likely in future iterations of the algorithm [5,6].

$$Q_{pointer}(t) = a * B_{bestsolution}(t) + (1 - a) * (1 - B_{bestsolution}(t)) \tag{3}$$

$$Q(t + 1) = Q_{pointer}(t) + b * randnorm(0, 1) \tag{4}$$

where $B_{bestsolution}(t)$ is the best solution found by iteration t. $Q_{pointer}(t)$ is a temporary quantum chromosome which is used to guide the generation of $Q(t+1)$ towards the form of $B_{bestsolution}$. The term $randnorm(0, 1)$ is a random number drawn from a (0,1) normal distribution. The parameters a and b control the balance between exploration and exploitation, with a governing the importance attached to $B_{bestsolution}(t)$ and b governing the degree of variance generation, centred on $Q_{pointer}(t)$. Values of $a \in [0.1, 0.5]$ and $b \in [0.05, 0.15]$ are suggested by [5,6].

3 Option Pricing Model Calibration

An optimisation problem in financial modelling is considered to test the performance of the QIGA. The optimisation involves calibrating an option pricing model to observed market data. Calibration is a method of choosing model parameters so that the distance between a set of model option prices and market option prices is minimised, where distance is some metric such as the sum of squared errors or the sum of squared percentage errors. The parameters can be thought to resemble the market's view on current option prices and the underlying asset price. In calibration we do not explicitly take into account any historical data. All necessary information is contained in today's option prices which can be observed in the market. Practitioners frequently calibrate option pricing models so that the models provides a reasonable fit to current observed market option prices and they then use these models to price exotic derivatives or for hedging purposes. In this paper we calibrate a popular extension of the Black-Scholes [16] option pricing model known as the Variance Gamma (VG) model [17,18,19] to FTSE 100 index option data.

A European call option on an asset S_t with maturity date T and strike price K is defined as a contingent claim with payoff at time T given by $\max [S_T - K, 0]$.

The well known Black-Scholes (BS) formula for the price of a call on this asset is given by

$$C_{BS}\left(S_t, K, r, q, \tau; \sigma\right) = S_t e^{-q\tau} N\left(d_1\right) - K e^{-r\tau} N\left(d_1\right)$$

$$d_1 = \frac{-\ln m + \left(r - q + \frac{1}{2}\sigma^2\right)\tau}{\sigma\sqrt{\tau}} \qquad d_2 = d_1 - \sigma\sqrt{\tau}$$

where $\tau = T - t$ is the time-to-maturity, t is the current time, $m = K/S$ is the moneyness of the option, r and q are the continuously compounded risk-free rate and dividend yield and $N(\cdot)$ is the cumulative normal distribution function. Suppose a market option price, denoted by $C_M\left(S_t, K\right)$, is observed. The Black-Scholes implied volatility for this option price is that value of volatility which equates the BS model price to the market option price as follows

$$\sigma_{BS}\left(S_t, K\right) > 0$$
$$C_{BS}\left(S_t, K, r, \tau; \sigma_{BS}\left(S_t, K\right)\right) = C_M\left(S_t, K\right)$$

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.

Many different option pricing models have been proposed as alternatives to the BS model. Examples include stochastic volatility models and jump diffusion models which allow for more complex asset price dynamics. We examine one such simple extension of the BS model known as the Variance Gamma (VG) option pricing model. The idea is to model stock price movements occurring on business time rather than on calendar time using a time transformation of a Brownian motion. The resulting model is a three parameter model where roughly speaking we can interpret the parameters as controlling volatility, skewness and kurtosis, denoted respectively as σ, θ and ν, of the underlying asset returns distribution. Closed form option pricing formulae exist under the VG model [19].

$$\begin{aligned} C_{VG}\left(S_t, K, r, \tau; \{\sigma, \nu, \theta\}\right) = & S_t e^{-q\tau} \Psi\left(d\sqrt{\frac{1-c_1}{\nu}}, \left(\alpha + s\right)\sqrt{\frac{\nu}{1-c_1}}, \frac{\tau}{\nu}\right) \\ & - K e^{-r\tau} \Psi\left(d\sqrt{\frac{1-c_2}{\nu}}, \alpha s\sqrt{\frac{\nu}{1-c_2}}, \frac{\tau}{\nu}\right) \end{aligned}$$

where

$$d = \frac{1}{s}\left[\ln\left(\frac{S_t}{K}\right) + \left(r - q\right)\tau + \frac{\tau}{\nu}\ln\left(\frac{1-c_1}{1-c_2}\right)\right]$$

$$\alpha = \varsigma s, \qquad \varsigma = -\frac{\theta}{\sigma^2}, \qquad s = \frac{\sigma}{\sqrt{1 + \left(\frac{\theta}{\sigma}\right)^2 \frac{\nu}{2}}}$$

$$c_1 = \frac{\nu\left(\alpha + s\right)^2}{2}, \qquad c_2 = \frac{\nu\alpha^2}{2}$$

and where Ψ is defined in terms of the modified Bessel function of the second kind.

Table 1. Market BS implied volatilities and option prices for FTSE 100 index options on the 17 March 2006. The strike prices are given in the table and the other observable inputs are $S = 5999.4, \tau = \frac{35}{365}, r = 0.0452$ and $q = 0.0306$.

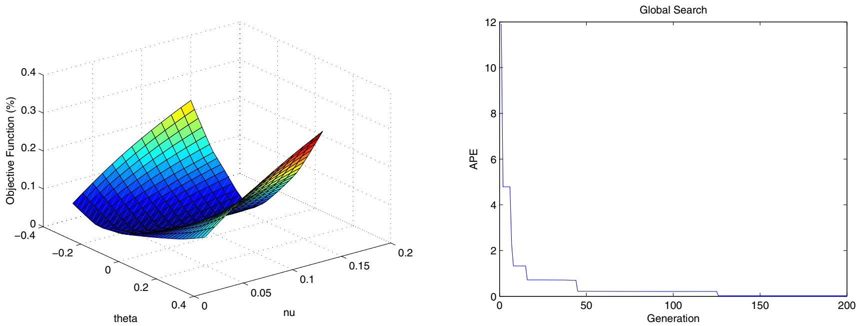
Strike price	5695.2	5845.1	5995.0	6144.9	6294.7
IV (%)	13.76	12.41	11.13	10.44	10.94
Call($)	323.67	193.63	88.67	28.03	7.99
Put ($)	12.44	31.63	75.89	164.48	293.67

4 Experimental Approach

Market makers in the options markets quote BS implied volatilities rather than option prices even though they realise BS is a flawed model. The first row in Table 1 depicts end-of-day settlement Black-Scholes implied volatilities for FTSE 100 European options on the 17 March 2006 for different strike prices and a time-to-maturity of 35 days. As can be seen the BS implied volatilities are not constant across the strike price. The second and third row in Table 1 converts the BS implied volatilities into market call and put prices by substituting the BS implied volatilities into the Black-Scholes formula. The following input parameters were used to calculate the option prices, the index price is the FTSE 100 index itself $S_t = 5999.4$, the interest rate is the one month Libor rate converted into a continuously compounded rate $r = 0.0452$ and the dividend yield is a continuously compounded dividend yield downloaded from Datastream and is $q = 0.0306$. These prices are then taken to be the observed market option prices. Out-of-the money (OTM) option prices are considered most suitable for calibration purposes because of their liquidity and informational content. Hence OTM put prices were used for $K < S$ and OTM call prices were used for $K > S$ in the calibration. The calibration problem now amounts to choosing an optimum parameter vector $\Theta = \{\sigma, \nu, \theta\}$ such that an objective function $G(\Theta)$ is minimised. In this paper the objective function is chosen to be the absolute average percentage error (APE)

$$\mathrm{G}(\Theta) = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{C_i - C_i(\Theta)}{C_i} \right|$$

where C_i is the observed market price on the i-th option (could be a call or a put) and $C_i(\Theta)$ is the VG model price of the i-th option with parameter vector Θ. One of the difficulties in model calibration is that the available market information may be insufficient to completely identify the parameters of a model [20]. If the model is sufficiently rich relative to the number of market prices available, a number of possible parameter vector combinations will be compatible with market prices and the objective function $G(\Theta)$ may not be convex function of Θ. A plot of the objective function versus the two parameters controlling skewness and kurtosis of the asset returns distribution, θ and ν, whilst keeping σ fixed at $\sigma = 0.1116$ is shown in figure 1(a).



(a) Objective function vs parameters (b) Objective function vs generation no.

Fig. 1. Objective function versus model parameters ν and θ and objective function versus generation number

It displays a flat profile near the minimum where many parameter combinations will yield equivalent fits. The error surface is not a straightforward error surface and a local optimiser might not converge to the true optimum. There are regions where the error surface is very flat for changes in the parameter values and there are regions where the optimiser might get not converge to global optimum.

5 Results

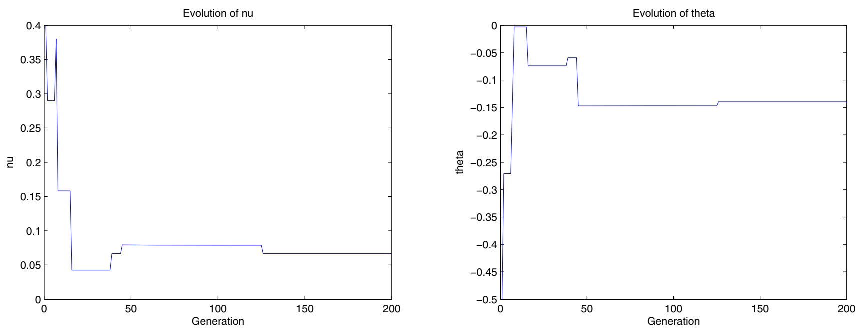
In all runs of the QIGA, a population size of 50 observed chromosomes was used, the algorithm was allowed to run for 200 generations, and all reported results are averaged over 30 runs. In order to provide a benchmark for the results obtained by the QIGA a deterministic Matlab optimiser called *fminsearch* was run 30 times with different initial parameter vectors. *Fminsearch* uses the simplex search method of [21]. This is a direct search method that does not use numerical or analytic gradients. The optimiser converged to different values for Θ for different

Table 2. Results of QIGA where the mean parameter values after 30 runs and the best performing parameter values are compared with the parameters from the matlab optimiser *fminsearch*. The resulting mean model prices from the 30 runs are compared with the market prices and the mean APE is reported.

Parameter	Mean QIGA	Best QIGA	Matlab	Market Price	Mean Model Price	Best Model Price
σ	0.0926	0.1055	0.1143	12.44	17.13	13.43
ν	0.3302	0.0234	0.0638	31.64	32.62	35.50
θ	-0.2316	-0.4258	-0.1429	75.90	65.66	83.13
				28.02	22.10	32.41
APE	2.5099	0.6000		7.99	7.03	6.75

Table 3. This table reports the QIGA objective function for different values of the exploitation and exploration parameters, respectively given by a and b

$a \setminus b$	0.4	2.0	3.6	5.2	6.8	8.4
0.05	0.5035	0.1808	0.4192	0.0411	0.0273	0.2265
0.25	0.5035	0.0791	0.4868	0.3789	0.1536	0.0992
0.45	0.5035	0.3560	0.2080	0.1984	0.0993	0.2870
0.65	0.5035	0.1298	0.1015	0.0179	0.0633	0.0907
0.85	0.5035	0.3385	0.5035	0.5035	0.5035	0.4114



(a) ν (b) θ

Fig. 2. Evolution of parameters ν and θ as a function of the generation number

initialisations of the parameter vector so the one with the optimal value for the objective function G was chosen. The results are reported in the Tables 2 and 3. As can be seen when averaged over only 30 runs the QIGA parameter vector Θ is reasonably close to the optimal parameter vector from matlab. Figure 1(b) depicts the evolution of the global objective function G (also known as APE) as a function of the generation number. Figures 2(a) and 2(b) depict the evolution of the parameters ν and θ as a function of the generation number. The results reported in Table 3 depicts the sensitivity of the objective function value to the exploitation and exploration parameters, respectively a and b. In this table a and b are varied to those values reported in while everything else remains fixed. It can be seen that the performance of the QIGA is not good when b is low regardless of what value a takes. As b increases more exploration takes place and the performance of the algorithm improves especially when a is set to intermediate values (approx. 0.65). Further sensitivity analysis would need to be conducted to find optimal values for these parameters.

6 Conclusions and Future Work

This study illustrates how a quantum-inspired evolutionary algorithm can be constructed and examines the utility of the resulting algorithm on a problem in financial modelling known as model calibration. The results from the algorithm are shown to be robust and comparable to those of other algorithms.

Several extensions of the methodology in this study are indicated for future work. Algorithms extensions would include developing and testing a real valued QIGA and comparing its performance to the binary algorithm used in this paper. Financial applications include the calibration of more complex higher dimensional option pricing models that may contain many local minima to market data in an evolutionary setting. The use of QIGA in these types of problems may be crucial due to the potential reduction in computational time.

References

1. Spector, L. (2004). *Automatic Quantum Computer Programming: A Genetic Programming Approach*, Boston, MA: Kluwer Academic Publishers.
2. Narayanan, A. and Moore, M. (1996). Quantum-inspired genetic algorithms, *Proceedings of IEEE International Conference on Evolutionary Computation*, May 1996, pp. 61-66, IEEE Press.
3. Han, K-H. and Kim, J-H. (2002). Quantum-inspired evolutionary algorithm for a class of combinatorial optimization, *IEEE Transactions on Evolutionary Computation*, 6(6):580-593.
4. Han, K-H. and Kim, J-H. (2002). Quantum-inspired evolutionary algorithms with a new termination criterion, H_ε gate and two-phase scheme, *IEEE Transactions on Evolutionary Computation*, 8(3):156-169.
5. Yang, S., Wang, M. and Jiao, L. (2004). A genetic algorithm based on quantum chromosome, *Proceedings of IEEE International Conference on Signal Processing* (ICSP 04), 31 Aug- 4 Sept. 2004, pp. 1622-1625, IEEE Press.
6. Yang, S., Wang, M. and Jiao, L. (2004). A novel quantum evolutionary algorithm and its application, *Proceedings of IEEE Congress on Evolutionary Computation 2004* (CEC 2004), 19-23 June 2004, pp. 820-826, IEEE Press.
7. Brabazon, A. and O'Neill, M. (2006). *Biologically-inspired Algorithms for Financial Modelling*, Berlin: Springer.
8. Yang, S., Wang, M. and Jiao, L. (2004). A Quantum Particle Swarm Optimization, in *Proceedings of the Congress on Evolutionary Computation 2004*, 1:320-324, New Jersey: IEEE Press.
9. Lee C-D., Chen, Y-J., Huang, H-C., Hwang, R-C. and Yu, G-R. (2004). The non-stationary signal prediction by using quantum NN, *Proceedings of 2004 IEEE International Conference on Systems, Man and Cybernetics*, 10-13 Oct. 2002, pp. 3291-3295, IEEE Press.
10. Garavaglia, S. (2002). A quantum-inspired self-organizing map (QISOM), *Proceedings of 2002 International Joint Conference on Neural Networks* (IJCNN 2002),12-17 May 2002, pp. 1779-1784, IEEE Press.
11. Tsai, X-Y., Chen, Y-J., Huang, H-C., Chuang, S-J. and Hwang, R-C. (2005). Quantum NN vs NN in Signal Recognition, *Proceedings of the Third International Conference on Information Technology and Applications* (ICITA 05), 4-7 July 2005, pp. 308-312, IEEE Press.
12. Li, Y., Zhang, Y., Zhao, R. and Jiao, L. (2004). The immune quantum-inspired evolutionary algorithm, *Proceedings of 2004 IEEE International Conference on Systems, Man and Cybernetics*, 10-13 Oct. 2002, pp. 3301-3305, IEEE Press.

13. Jiao, L. and Li, Y. (2005). Quantum-inspired immune clonal optimization, *Proceedings of 2005 International Conference on Neural Networks and Brain* (ICNN&B 2005), 13-15 Oct. 2005, pp. 461-466, IEEE Press.
14. da Cruz, A, Vellasco, M. and Pacheco, M. (2006). Quantum-inspired evolutionary algorithm for numerical optimization, in *Proceedings of the 2006 IEEE Congress on Evolutionary Computation (CEC 2006)*, 16-21 July, Vancouver, pp. 9180-9187, IEEE Press.
15. Han, K-H. and Kim, J-H. (2003). On setting the parameters of quantum-inspired evolutionary algorithm for practical applications, *Proceedings of IEEE Congress on Evolutionary Computing* (CEC 2003), 8 Aug-12 Dec. 2003, pp. 178-184, IEEE Press.
16. Black, F. and Scholes, M. (1973). The pricing of options and corporate liabilities, *Journal of Political Economy*, 81, pp. 637-654.
17. Madan, D. and Seneta, E. (1990). The VG model for share market returns, *Journal of Business*, 63, pp. 511 - 524.
18. Madan, D. and Milne, F. (1991). Option pricing with VG martingale components, *Mathematical Finance*, 1 (4), pp. 39 - 55.
19. Madan, D., Carr, P. and Chang, E. (1998). The Variance Gamma Process and Option Pricing, *European Finance Review*, 2, pp. 79 - 105.
20. Cont, R. and Ben Hamida, S. (2005). Recovering volatility from option prices by evolutionary optimisation, *Journal of Computational Finance*, 8 (4), Summer 2005.
21. Lagarias, J.C., J. A. Reeds, M. H. Wright, and P. E. Wright. (1998) .Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions, *SIAM Journal of Optimization*, Vol. 9, Number 1, pp.112-147, 1998.