

# Option Pricing Model Calibration using a Real-valued Quantum-inspired Evolutionary Algorithm

Kai Fan  
Natural Computing Research  
and Applications Group  
University College Dublin  
kai.fan@ucd.ie

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

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

Michael O'Neill  
Natural Computing Research  
and Applications Group,  
University College Dublin  
m.oneill@ucd.ie

## ABSTRACT

Quantum effects are a natural phenomenon and just like evolution, or immune processes, can serve as an inspiration for the design of computing algorithms. This study illustrates how a real-valued quantum-inspired evolutionary algorithm (QEA) can be constructed and examines the utility of the resulting algorithm on an important real-world problem, namely the calibration of an Option Pricing model. The results from the algorithm are shown to be robust and sensitivity analysis is carried out on the algorithm parameters, suggesting that there is useful potential to apply QEA to this domain.

**Categories and Subject Descriptors:** Real-World Applications - Finance

**General Terms:** Experimentation

**Keywords:** Real-valued quantum-inspired evolutionary algorithm, option pricing

## 1. INTRODUCTION

This study applies a quantum evolutionary algorithm in order to calibrate a popular option pricing model used extensively in industry known as the Variance Gamma option pricing model.

In recent years there has been a substantial interest in the theory and design of quantum computers and in 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 [18]. 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 [16, 6, 8, 20, 21], social computing [22], neuro-computing [11, 5, 19], and immunocomputing [12, 9]. 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.

While quantum-inspired evolutionary algorithms (QEA) offer interesting potential, as yet due to their novelty, only a small number of recent papers have implemented a QEA, typically reporting good results [20, 21]. 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 [2], the QEA methodology has not yet been applied to the finance domain. This study addresses both of these research 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) [16, 20, 21]. 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

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'07, July 7–11, 2007, London, England, United Kingdom.  
Copyright 2007 ACM 978-1-59593-697-4/07/0007 ...\$5.00.

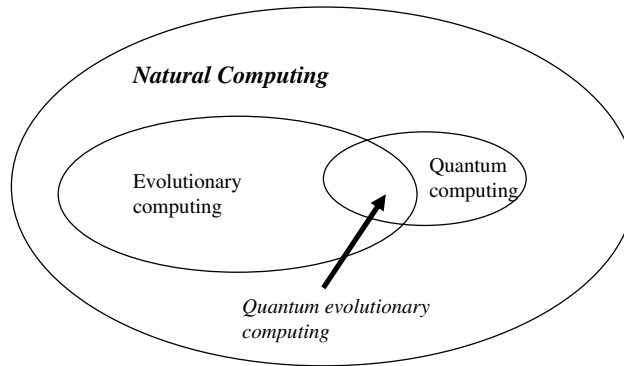


Figure 1: Quantum-inspired evolutionary computing

$$|q^i\rangle = \alpha|0\rangle + \beta|1\rangle \quad (1)$$

where  $|0\rangle$  and  $|1\rangle$  are the ground states 0 and 1, and  $\alpha$  &  $\beta$  are complex numbers with  $|\alpha|^2 + |\beta|^2 = 1$ , 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  $\alpha$  and  $\beta$  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

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_m \\ \beta_1 & \beta_2 & \dots & \beta_m \end{bmatrix} \quad (2)$$

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 (in the limit) 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 Real-valued quantum-inspired evolutionary algorithms

In the initial literature which introduced the QIGA, a binary representation was adopted, wherein each quantum chromosome was restricted to consist of a series of 0s and 1s. The methodology was recently extended to include real-valued vectors by da Cruz et al. [4]. As for binary-representation QIGA, real-valued QIGA maintains a distinction between a quantum population and an observed population of (in this case) real-valued solution vectors. However the quantum individuals have a different form to those in binary-representation QIGA. The quantum population  $Q(t)$  is comprised of  $N$  quantum individuals ( $q_i : i = 1, 2, 3, \dots, N$ ), where each individual  $i$  is comprised of  $G$  genes ( $g_{ij} : j = 1, 2, 3, \dots, G$ ). Each of these genes consist of a pair of values  $q_{ij} = (p_{ij}, \sigma_{ij})$  where  $p_{ij}, \sigma_{ij} \in \mathfrak{R}$  represent the mean and the width of a square pulse. Representing a gene in this manner has a parallel with the quantum concept of superimposition of states as a gene is specified by a range of possible values, rather than by a single unique value.

### 2.1.1 Algorithm

The real-valued QEA algorithm is as follows

```

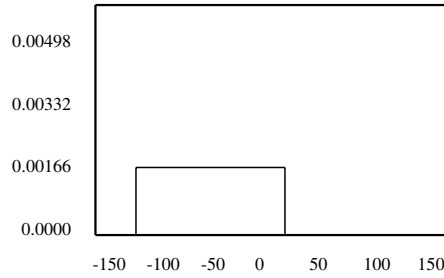
Set t=0

Initialise Q(t) of N individuals with G genes

While (t < max t)
  Create the PDFs (and corresponding CDFs) for each gene locus
  using the quantum individuals
  Create a temporary population, denoted E(T), of K real-valued
  solution vectors by observing
  Q(t) (via the CDFs)
  If (t=1) Then C(t)=E(t)
  (Note: the population C(T) is maintained between iterations of
  the algorithm)
  Else
    E(t)=Outcome of crossover between E(t) and C(t)
    Evaluate E(t)
    C(t)= K best individuals from E(t) U C(t)
  End if

  With the N best individuals from C(t)
  Q(t+1)=Output of translate operation on Q(t)
  Q(t+1)=Output of resize operation on Q(t+1)
  t=t+1
Endwhile

```



**Figure 2: A square pulse, representing a quantum gene, with a width of 150, centred on -50. The height of the gate is 0.00166**

### 2.1.2 Initialising the Quantum Population

At the start of the algorithm, the quantum genes in each member of the quantum population are initialised by randomly selecting a value from within the range of allowable values for that dimension. A gene's width value is set to the range of allowable values for the dimension. For example, if the known allowable values for dimension  $j$  are  $[-75, 75]$  then  $q_{ij}$  is initially determined by randomly selecting a value from this range (say) -50, and the corresponding width value will be 150. Hence,  $q_{ij} = (-50, 150)$ . The square pulse need not be entirely within the allowable range for a dimension when it is initially created as the algorithm will automatically adjust for this as it executes. The height of the pulse arising from a gene is calculated using

$$h_{ij} = \frac{1/\sigma_{ij}}{N} \quad (3)$$

where  $N$  is the number of individuals in the quantum population. This equation ensures that the probability density functions (PDFs) (see next subsection) used to generate the observed individual solution vectors will have a total area equal to one. Fig. 2 provides an illustration of a quantum gene where  $N=4$ .

### 2.1.3 Observing the Quantum Chromosomes

In order to generate a population of real-valued solution vectors, a series of observations must be undertaken using the quantum population. A pseudo-interference process between the quantum individuals is simulated by summing up the square pulses for each individual gene across all members of the quantum population. This generates a separate PDF (just the sum of the square pulses) for each gene and eq. 3 ensures that the area under this PDF is one. Hence, the PDF for gene  $j$  on iteration  $t$  is

$$PDF_j(t) = \sum_i^j g_{ij} \quad (4)$$

where  $g_{ij}$  is the square pulse of the  $j^{th}$  gene of the  $i^{th}$  quantum individual (of  $N$ ). To use this information to obtain an observation, the PDF is first converted into its corresponding Cumulative Distribution Function (CDF)

$$CDF_j(x) = \int_{L_j}^{U_j} PDF_j(x) dx \quad (5)$$

where  $U_j$  and  $L_j$  are the upper and lower limits of the probability distribution. By generating a random number  $r$  from  $(0,1)$ , the CDF can be used to obtain an observation of a real number  $x$ , where  $x = CDF^{-1}(r)$ . If the generated value  $x$  is outside the allowable real valued range for that dimension, the generated value is limited to its allowable boundary value. A separate PDF and CDF is calculated for each of the  $G$  gene positions. Once these have been calculated, the observation process is iterated to create a temporary population with  $K$  members, denoted  $E(t)$ .

### 2.1.4 Crossover Mechanism

The crossover operation takes place between  $C(t)$  and the temporary population  $E(t)$ . This step could be operationalised in a variety of ways with [4] choosing to adopt a variant of uniform crossover, without an explicit selection operator. After the  $K$  crossover operations have been performed, with the resulting children being copied into  $E(t)$ , the best  $K$  individuals  $\in C(t) \cup E(t)$  are copied into  $C(t)$ .

### 2.1.5 Updating the Quantum Chromosomes

The  $N$  quantum chromosomes are updated using the  $N$  best individuals from  $C(t)$  after performing the crossover step. Each quantum gene's mean value is altered using

$$p_{ij} = c_{ij} \quad (6)$$

so that the mean value of the  $j^{th}$  gene of the  $i^{th}$  quantum chromosome is given by the corresponding  $j^{th}$  value of the  $i^{th}$  ranked individual in  $C(t)$ .

The next step is to update the corresponding width value of the  $j^{th}$  gene. The objective of this process is to vary the exploration / exploitation characteristics of the search algorithm, depending on the feedback from previous iterations. If the search process is continuing to uncover many new better solutions, then the exploration phase should be continued by keeping the widths relatively broad. However, if the search process is not uncovering many new better solutions, the widths are reduced in order to encourage finer-grained search around already discovered good regions of the solution space. In this paper we update the width of the  $i^{th}$  quantum chromosome's  $j^{th}$  gene by comparing each successive generations best fitness function. If the best fitness function has improved (disimproved) we shrink (enlarge) the width in order to improve the local (global) search.

## 2.2 QIGA vs Canonical Genetic Algorithm

A number of distinctions between the QIGA above and the canonical GA (CGA) can be noted. In the CGA, the population of solutions persists from generation to generation, albeit in a changing form. In contrast, in QIGA, the population of solutions in  $P(t)$  are discarded at the end of each loop. The described QIGA, unlike CGA, does not have explicit concepts of crossover or mutation. However, the adaptation of the quantum chromosomes in each iteration does embed implicit selection as the best solution is selected and is used to adapt the quantum chromosome(s). The crossover and mutation steps are also implicitly present, as the adaptation of the quantum chromosome in effect creates diversity, as it makes different states of the system more or less likely over time. Another distinction between the QIGA and the CGA is that the CGA operates directly on representations of the solution (the members of the current population of solutions), whereas in QIGA the update step is performed on the probability amplitudes of the ground states for each qubit making up the quantum chromosome(s).

The real-valued QIGA is a novel form of representation which could be hybridised with multiple real-valued search algorithms apart from the GA such as PSO or DE. The methodology has shown promise on the limited range of test problems it has been applied to. However, further testing is necessary in order to better assess its computational efficiency and its overall effectiveness on a wider range of problems. Future work could also assess the benefits of operationalising the key steps of the algorithm in alternative ways, for example, by implementing alternative diversity-generating mechanisms in updating  $C(t)$  or by implementing alternative mechanisms for dynamically altering  $\sigma$  as the algorithm runs.

## 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 [1] option pricing model known as the Variance Gamma (VG) model [13, 14, 15] 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

$$C_{BS}(S_t, K, r, q, \tau; \sigma) = S_t e^{-q\tau} N(d_1) - K e^{-r\tau} N(d_2)$$

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

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  $C_M(S_t, K)$ , 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

$$\sigma_{BS}(S_t, K) > 0$$

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

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 reality 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 (as assumed by BS), and take into account the resulting complications, this result begins to make sense and is simply highlighting the erroneous assumptions that underpin the BS model.

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  $\sigma, \theta$  and  $\nu$ , of the underlying asset returns distribution. Closed form option pricing formulae exist under the VG model [15].

$$C_{VG}(S_t, K, r, \tau; \{\sigma, \nu, \theta\}) =$$

$$S_t e^{-q\tau} \Psi\left(d\sqrt{\frac{1-c_1}{\nu}}, (\alpha + s)\sqrt{\frac{\nu}{1-c_1}}, \frac{\tau}{\nu}\right)$$

$$- K e^{-r\tau} \Psi\left(d\sqrt{\frac{1-c_2}{\nu}}, \alpha s\sqrt{\frac{\nu}{1-c_2}}, \frac{\tau}{\nu}\right)$$

where

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

$$\alpha = \zeta s, \quad \zeta = -\frac{\theta}{\sigma^2}, \quad s = \frac{\sigma}{\sqrt{1 + \left(\frac{\theta}{\sigma}\right)^2 \frac{\tau}{2}}}$$

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

and where  $\Psi$  is defined in terms of the modified Bessel function of the second kind.<sup>1</sup>

#### 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 call prices were used for  $K < S$  and OTM put 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)

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

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  $\Theta$ . One of the difficulties in model calibration is that the available market information may be insufficient to completely identify the parameters of a model [3]. 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  $\Theta$ . A plot of the objective function versus the two parameters controlling skewness and kurtosis of the asset returns distribution,  $\theta$  and  $\nu$ , whilst keeping  $\sigma$  fixed at  $\sigma = 0.1116$  is shown in figure 3(a).

It displays a very flat profile near the minimum where many parameter combinations will yield equivalent fits i.e. different parameter combinations will have almost identical APE values. This is because the two parameters  $\theta$  and  $\nu$  have slightly offsetting effects. Cont and Tankov [3] also graph a similar objective function for the  $VG$  option pricing model using DAX index option data and illustrate the potential for gradient based optimisers to converge to a local rather than the global minimum.

#### 5. RESULTS

In all runs of the QIGA, a population size of 20 observed chromosomes was used, the algorithm was allowed to run for 30 generations, and all reported results are averaged over 30

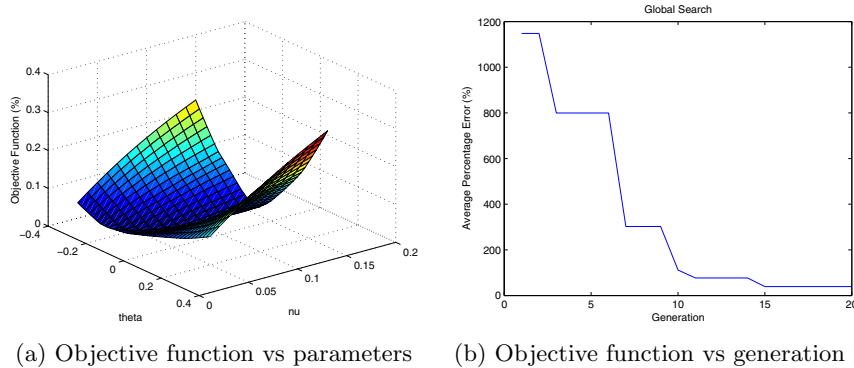
runs. In order to provide a benchmark for the results obtained by the QIGA a deterministic Matlab optimiser called *fminsearch* was run 30 times with different initial parameter vectors. *Fminsearch* uses the simplex search method of Lagarias et al [10]. This is a direct search method that does not use numerical or analytic gradients. The optimiser converged to different values for the parameter vector  $\Theta$  for different initialisations of  $\Theta$ . The parameter vector associated with the optimal value for the objective function  $G$  was chosen to compute the average percentage error (APE) and the model prices. The optimal results from the last generation of the algorithm are averaged over 30 runs and are reported in Table 2. The average results from the last generation of the algorithm are averaged over 30 runs and are reported in Table 3. As can be seen in Table 2 the optimal APE is quite low at 7.93% and the parameter values are very close to the parameter values from the Matlab optimiser. The results from Table 3 average fitness results. The APE is calculated by computing a single set of model prices from the average parameter vector  $\Theta$ . The APE is higher than in the previous table at 14.87% indicating that some of the runs converged too early to a suboptimal result. However the average parameter values are not that different from the optimal parameter values indicating that only a small number of the 30 runs gave poor results. Figure 3(b) depicts the evolution of the global objective function  $G$  (also known as the average percentage error (APE)) as a function of the generation number for a single run of the algorithm. Figures 4(a) and 4(b) depict the evolution of the parameters  $\nu$  and  $\theta$  as a function of the generation number for a single run of the algorithm.

In previous expositions of the real-valued QIGA, detailed sensitivity analysis results were not reported. In order to gain greater insight into the operation of the algorithm, and to guide future applications of it, we undertook such an analysis by systematically investigating a variety of parameter settings for shrinkage, enlargement and crossover. The results are reported in tables 4 and 5. The optimal APE value as a function of the enlargement and shrinkage parameters is reported in table 4 reports. The crossover rate is fixed at 0.5, a population size of 20 and a generation number of 20 are used. Figure 5(a) graphs these results. The APE is less sensitive to the enlargement parameter than to the shrinkage parameter. The shrinkage parameter forces the algorithm to converge faster and this has a strong effect on the algorithms performance. The enlargement parameter causes the algorithm to widen the search space, however the crossover rate can also do this albeit using a different method, and this is why the algorithm is less sensitive to the enlargement parameter provided the crossover rate is a reasonable value. In this paper the shrinkage parameter is set to 0.8 and the enlargement parameter is set to 1.2 as this provided a reasonable trade-off in the convergence speed of the algorithm versus the search space of the algorithm. Table 5 reports the optimal APE value as a function of the shrinkage parameter and crossover rate. The enlargement parameter is set to 1.2 and a population size of 20 and generation number of 20 are used. Figure 5(b) graphs these results. The APE is less sensitive to the crossover rate than to the shrinkage parameter when the enlargement is set to 1.2. For low and high values of the crossover rate the APE is large and is lowest when the crossover rate is approximately 0.5. Low values of the crossover rate results in a small search space and high

<sup>1</sup>See [15] for the exact form of this function.

**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



**Figure 3: Objective function versus model parameters  $\nu$  and  $\theta$  and objective function versus generation number.**

**Table 2: Results of QIGA. The optimal parameter values from the last generation are averaged over 30 runs and compared with the parameter values from 30 runs of a Matlab optimiser. The corresponding APE and model prices are reported (also averaged over 30 runs) along with market option prices.**

Parameter	QIGA	Matlab	Market Price	Optimal Model Price	Optimal APE (%)
mean $\sigma$	0.1144	0.1143	12.44	12.39	7.93
mean $\nu$	0.0724	0.0638	31.64	31.79	
mean $\theta$	-0.1257	-0.1429	75.90	76.85	
			28.02	30.98	
			7.99	9.85	

**Table 3: Results of QIGA. The average parameter values from the last generation are averaged over 30 runs and compared with the parameter values from 30 runs of a Matlab optimiser. The corresponding APE and model prices are reported (also averaged over 30 runs) along with market option prices.**

Parameter	QIGA	Matlab	Market Price	Mean Model Price	Mean APE (%)
mean $\sigma$	0.1148	0.1143	12.44	12.31	14.87
mean $\nu$	0.0686	0.0638	31.64	31.55	
mean $\theta$	-0.1464	-0.1429	75.90	76.19	
			28.02	29.40	
			7.99	8.67	

**Table 4: This table reports the APE (%) for different enlargement and shrinkage values.**

Enlargement\Shrinkage	0.5	0.6	0.7	0.8	0.9
1.1	163.92	132.14	43.83	27.42	35.88
1.2	187.53	115.29	32.48	16.48	41.80
1.3	122.74	159.21	28.06	28.91	31.58
1.4	165.91	98.31	39.35	66.93	78.94
1.5	172.40	135.55	94.75	53.34	63.32

values results in almost random search so intermediate values provide a reasonable trade-off between exploitation and exploration. In this paper the crossover rate is set to 0.5.

## 6. CONCLUSIONS

This study illustrates how a real valued quantum-inspired evolutionary algorithm (QIGAs) can be constructed and examines the utility of the resulting algorithm on a financial model calibration problem. The results are shown to be robust and comparable to those of other algorithms. This underpins earlier proof of concept exploration studies using real valued quantum-inspired evolutionary algorithm. It is also noted that this paper reports the first application of a QIGA to the financial domain. Several extensions of the methodology in this study are indicated for future work. The first extension would be to extend the real-valued QIGA to a higher dimensional setting so that the computational benefits of the real-valued QIGA really begin to take affect. The algorithm offers substantial potential for calibrating more complex financial models than the *VG* option pricing model. Other extensions include the examination of the utility of other forms of QIGAs, such as binary valued QIGAs.

## 7. REFERENCES

- [1] Black, F. and Scholes, M. (1973). The pricing of options and corporate liabilities, *Journal of Political Economy*, 81:37-654.
- [2] Brabazon, A. and O'Neill, M. (2006). *Biologically-inspired Algorithms for Financial Modelling*, Berlin: Springer.
- [3] Cont, R. and Ben Hamida, S. (2005). Recovering volatility from option prices by evolutionary optimisation, *Journal of Computational Finance*, 8 (4), Summer 2005.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] 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)*, pp. 178-184, IEEE Press.
- [8] Han, K-H. and Kim, J-H. (2002). Quantum-inspired evolutionary algorithms with a new termination criterion,  $H_\varepsilon$  gate and two-phase scheme, *IEEE Transactions on Evolutionary Computation*, 8(3):156-169.
- [9] 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.
- [10] 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.
- [11] 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.
- [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] Madan, D. and Seneta, E. (1990). The VG model for share market returns, *Journal of Business*, 63:511-524.
- [14] Madan, D. and Milne, F. (1991). Option pricing with VG martingale components, *Mathematical Finance*, 1(4):39-55.
- [15] Madan, D., Carr, P. and Chang, E. (1998). The Variance Gamma Process and Option Pricing, *European Finance Review*, 2:79-105.
- [16] 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.
- [17] Rechenberg, I. (1973). *Evolutionstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*, Fromman-Holzboog Verlag, Stuttgart.
- [18] Spector, L. (2004). *Automatic Quantum Computer Programming: A Genetic Programming Approach*, Boston, MA: Kluwer Academic Publishers.
- [19] 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.
- [20] 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.
- [21] 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.
- [22] 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.

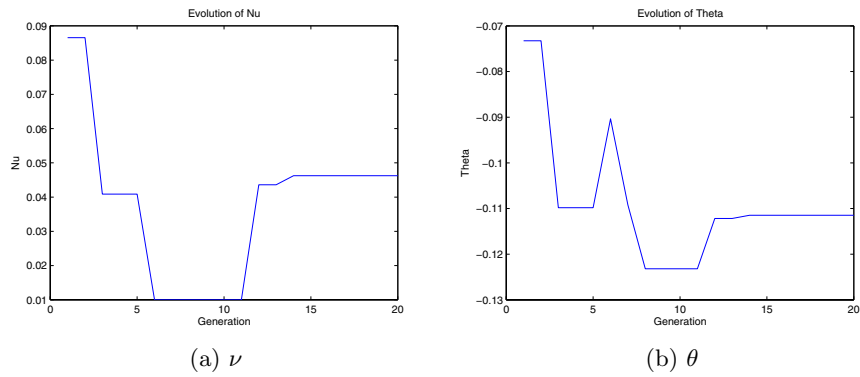


Figure 4: Evolution of parameters  $\nu$  and  $\theta$  over time.

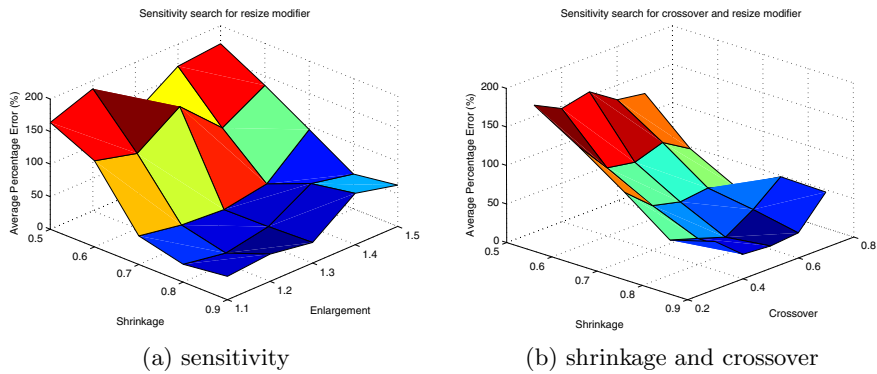


Figure 5: Sensitivity search

Table 5: This table reports the APE (%) for different enlargement and crossover values.

Crossover\Shrinkage	0.5	0.6	0.7	0.8	0.9
0.3	164.72	128.76	88.60	45.35	61.91
0.4	146.77	88.37	57.95	33.94	31.79
0.5	154.88	82.10	49.21	21.98	29.03
0.6	131.11	94.21	53.67	44.28	32.45
0.7	125.63	72.57	42.10	73.32	71.87