# Quantum-Inspired Evolutionary Algorithms for Financial Data Analysis

Kai Fan[1,2], 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.** This paper describes a real-valued quantum-inspired evolutionary algorithm (QIEA), a new computational approach which bears similarity with estimation of distribution algorithms (EDAs). The study assesses the performance of the QIEA on a series of benchmark problems and compares the results with those from a canonical genetic algorithm. Furthermore, we apply QIEA to a finance problem, namely non-linear principal component analysis of implied volatilities. The results from the algorithm are shown to be robust and they suggest potential for useful application of the QIEA to high-dimensional optimization problems in finance.

## 1 Introduction

A wide-variety of biologically-inspired algorithms have been applied for financial modelling [1] in recent years. One interesting avenue of this research has been the hybridisation of quantum-inspired concepts with evolutionary algorithms [11,8,13] producing a family of algorithms known as quantum-inspired evolutionary algorithms (QIEA). 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. As yet, apart from [4,5,6], there have been no studies applying these algorithms in the finance domain. This paper extends these proof of concept studies in two important ways. First, it explores the utility of a real-valued QIEA by applying the algorithm to a series of benchmark problems and comparing the results with those produced by a canonical GA. Then it applies the methodology to undertake a non-linear principal component analysis (NLPCA). The NLPCA is used to determine the non-linear principal components that drive the variations in the implied volatility smile for financial options.

The next section provides a short introduction to the real-valued QIEA and outlines the benchmark tests undertaken. This is followed by a description of the experimental approach adopted in the NLCPA, followed by the results and finally, the conclusions of this paper.

M. Giacobini et al. (Eds.): EvoWorkshops 2008, LNCS 4974, pp. 133–143, 2008.

## 2 Quantum-Inspired Evolutionary Algorithm

The real-valued quantum-inspired evolutionary algorithm applied in this paper is described in [5,6] and readers are referred there for more details. A short overview of the algorithm is provided below.

Quantum mechanics is an extension of classical mechanics which models behaviours of natural systems that are observed particularly at very short time or distance scales. In the initial literature which introduced the QIEA, a binary representation was adopted, wherein each quantum chromosome was restricted to consist of a series of 0s and 1s. The methodology was modified to include real-valued vectors by da Cruz et al., [3]. As with binary-representation QIEA, real-valued QIEA 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 QIEA. The quantum population $Q(t)$ is comprised of $N$ quantum individuals $(q_i : i = 1, 2, 3, \ldots, N)$, where each individual is comprised of $G$ genes $(g_{ij} : j = 1, 2, 3, \ldots, G)$. Each of these genes consist of a pair of values $g_{ij} = (p_{ij}, \sigma_{ij})$ where $p_{ij}, \sigma_{ij} \in \Re$ represent the mean and the width of a square pulse. Representing a gene in this manner has a parallel with the quantum concept of superposition of states as a gene is specified by a range of possible values, rather than by a single unique value.

The da Cruz et al algorithm does periodically sample from a distribution to get a "classical" population, which can be regarded as a wave-function (quantum state) collapsing to a classical state upon observation. It is also noted that this bears similarity to the operation of a number of estimation of distribution algorithms (EDAs).

**Algorithm.** The real-valued QIEA algorithm is as follows

**Algorithm.** Real-valued Quantum-inspired Genetic Algorithm

```
Set t=0;
Initalise Q(t) (the quantum chromosome);

while t < max_t do
    Create the PDFs (and corresponding CDFs) for each gene locus using the quantum
    individual;
    Create a temporary population, denoted E(T), of K real-valued solution vectors through
    a series of 'observations' via the CDFs;
    if t=0 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-1) and C(t-1);
        Evaluate E(t);
        C(t)= K best individuals from E(t) ∪ C(t-1);
    end
    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;
end
```

**Initialising the Quantum Population.** At the start of the algorithm, each quantum gene is initialised by randomly selecting a value from within the range of allowable values for that dimension. For example, if the known allowable values for dimension $j$ are $[-75, 75]$ then $q_{ij}$ (dimension $j$ in quantum chromosome $i$) is initially determined by randomly selecting a value from this range (say) -50. 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 $j$ in chromosome $i$ is calculated using

$$h_{ij} = \frac{1/\sigma_{ij}}{N} \tag{1}$$

where $N$ is the number of individuals in the quantum population. This equation ensures that the probability density functions (PDFs) used to generate the observed individual solution vectors will have a total area equal to one.

**Observing the Quantum Chromosomes.** In order to generate a population of real-valued solution vectors, a series of observations must be undertaken using the population of quantum chromosomes (individuals). 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. 1 ensures that the area under this PDF is one. Hence, the PDF for gene $j$ on iteration $t$ is

$$PDF_j(t) = \sum_{i}^{j} g_{ij} \tag{2}$$

where $g_{ij}$ is the squared 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)

$$CDF_j(x) = \int_{L_j}^{U_j} PDF_j(x)dx \tag{3}$$

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) following a specific distribution, the CDF can be used to obtain an observation of a real number $x$, where $x = CDF^{-1}(r)$. Once these have been calculated, the observation process is iterated to create a temporary population with $K$ members, denoted by E(t).

**Updating the Quantum Chromosomes.** In this study we adjust the quantum probability amplitude with specified operators by comparing each successive generation's best fitness function so that the quantum chromosome can produce more promising individuals with higher probability in the next generation, i.e. if the best fitness function has improved (disimproved) we shrink (enlarge) the width in order to improve the local (global) search.

**Table 1.** Benchmark functions

| $f$ | Function | Mathematical representation | Range | $f(x_i^*)$ | $x_i^*$ |
|---|---|---|---|---|---|
| $f_1$ | DeJong(Sphere) | $f(x) = \sum_{i=1}^{p} {x_i}^2$ | $-5 \leq x_i \leq 5$ | 0 | 0 |
| $f_2$ | Rosenbrock | $f(x) = \sum_{i=1}^{p-1} 100({x_{i+1}} - {x_i}^2)^2 + (1 - x_i)^2$ | $-50 \leq x_i \leq 50$ | 0 | 1 |
| $f_3$ | Rastrigin | $f(x) = 10p + \sum_{i=1}^{p} ({x_i}^2 - 10cos(2\pi x_i))$ | $-100 \leq x_i \leq 100$ | 0 | 0 |
| $f_4$ | Griewangk | $f(x) = \sum_{i=1}^{p} \frac{x_i^2}{4000} - \prod_{i=1}^{p} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$ | $-600 \leq x_i \leq 600$ | 0 | 0 |

**Table 2.** Parameters setting in QIEA and GA

| | | | | |
|---|---|---|---|---|
| QIEA | Population=50 | Observation=200 | Shrinkage=0.005 | Enlargement=30 |
| GA | Population=50 | Generations=200 | Mutation=0.005 | Crossover=0.75 |

## 3 Benchmark Test

Four major static benchmark functions are chosen to test the ability of QIEA to find a global minimum. The results are compared to those of a canonical GA. Details of the benchmark functions are shown in Table 1.

In order to make a fair comparison between QIEA and GA, we call the evaluation function 10000 times for both algorithms. The parameters used for QIEA and GA, selected from sensitivity test, are shown in Table 2.

**Table 3.** DeJong results

| [-5,5] | Best | Mean | S.D. | Time(s) |
|---|---|---|---|---|
| *Dimension* : 5 | | | | |
| QIEA | 0.0021 | 0.0012 | 0.0007 | 27.96 |
| GA | 0.0001 | 0.0000 | 0.0002 | 23.58 |
| *Dimension* : 10 | | | | |
| QIEA | 0.0378 | 0.0090 | 0.0216 | 28.28 |
| GA | 0.0002 | 0.0000 | 0.0002 | 39.74 |
| *Dimension* : 50 | | | | |
| QIEA | 3.427 | 0.289 | 2.562 | 29.21 |
| GA | 3.511 | 0.787 | 2.304 | 103.00 |
| *Dimension* : 100 | | | | |
| QIEA | 24.953 | 2.077 | 21.434 | 31.86 |
| GA | 54.778 | 6.809 | 39.564 | 190.63 |

**Table 4.** Rosenbrock results

| [-50,50] | Best | Mean | S.D. | Time(s) |
|---|---|---|---|---|
| *Dimension* : 5 | | | | |
| QIEA | 68.9335 | 44.1549 | 14.1392 | 29.36 |
| GA | 578.2585 | 1141.90 | 4.2406 | 24.34 |
| *Dimension* : 10 | | | | |
| QIEA | 786.7354 | 362.0550 | 80.5617 | 30.66 |
| GA | 939.3839 | 1334.200 | 10.8133 | 45.14 |
| *Dimension* : 50 | | | | |
| QIEA | 8.751e+5 | 2.245e+5 | 3.901e+5 | 31.04 |
| GA | 2.089e+6 | 7.242e+5 | 9.400e+5 | 125.34 |
| *Dimension* : 100 | | | | |
| QIEA | 3.50e+7 | 8.47e+6 | 1.88e+7 | 35.87 |
| GA | 1.56e+8 | 2.86e+7 | 1.07e+8 | 228.07 |

**Table 5.** Rastrigin results

| [-100,100] | Best | Mean | S.D. | Time(s) |
|---|---|---|---|---|
| *Dimension* : 5 | | | | |
| QIEA | 18.687 | 4.579 | 7.285 | 30.61 |
| GA | 15.777 | 6.016 | 9.186 | 21.21 |
| *Dimension* : 10 | | | | |
| QIEA | 47.485 | 10.104 | 25.542 | 30.97 |
| GA | 32.161 | 5.858 | 19.622 | 37.52 |
| *Dimension* : 50 | | | | |
| QIEA | 1479.3 | 190.5 | 866.9 | 37.84 |
| GA | 1763.8 | 276.6 | 1286.4 | 107.18 |
| *Dimension* : 100 | | | | |
| QIEA | 10599.0 | 858.7 | 8244.5 | 38.13 |
| GA | 22845.0 | 2594.9 | 18547.0 | 194.28 |

**Table 6.** Griewangk results

| [-600,600] | Best | Mean | S.D. | Time(s) |
|---|---|---|---|---|
| *Dimension* : 5 | | | | |
| QIEA | 0.398 | 0.092 | 0.182 | 44.64 |
| GA | 0.355 | 0.167 | 0.105 | 25.95 |
| *Dimension* : 10 | | | | |
| QIEA | 0.803 | 0.078 | 0.638 | 45.18 |
| GA | 0.397 | 0.107 | 0.214 | 25.95 |
| *Dimension* : 50 | | | | |
| QIEA | 10.06 | 1.35 | 5.65 | 47.92 |
| GA | 12.57 | 2.21 | 7.34 | 127.72 |
| *Dimension* : 100 | | | | |
| QIEA | 173.36 | 21.76 | 131.43 | 55.13 |
| GA | 204.37 | 24.80 | 163.53 | 232.33 |

The results of benchmark tests are shown in Tables 3 to 6, where both QIEA and GA are run 30 times. The first column lists the minimal (overall best) objective value found during the 30 runs. The second and third column lists the mean and standard deviation for the best value found in each of the 30 runs. The *Time* column shows the total processing time taken for 30 runs.

The results indicate that QIEA performs better (relative to GA) as the search space becomes more complex and the dimensionality of the search space becomes larger. It is also notable that the algorithm's efficiency vs the GA increases as the problem becomes larger. These results suggest the interesting potential of QIEA as an optimising algorithm in hard, high-dimensional problems.

## 4 Experimental Approach

In this section we explain the importance of the implied volatility smile in option trading and how QIEA will be applied for non-linear principal component analysis.

### 4.1 Implied Volatility Smile

Implied volatilities are frequently used in the financial markets to quote the prices of options. Option traders and brokers monitor movements in volatility smiles closely. As option prices change over time the implied volatility smile (for various maturities) also changes.

If we stack the implied volatility smile (for one particular maturity) according to the time the IVS data was recorded, a time series of panel data, with highly correlated entries, results. Implied volatilities at different strike prices are highly correlated because as the volatility of the asset rises all implied volatilities rise, yet some may rise more than others. However the economic forces of no-arbitrage (no free-lunch) ensures that the implied volatilities cannot get too detached from one another because if they did this represents a riskless trading opportunity for savvy investors, who sell the more expensive option (with the higher implied volatility) and hedge it with cheaper options (with lower implied volatilities).

### 4.2 Non-linear Principal Component Analysis

Suppose $X \in M^{m,n}$ is a panel data set that contains correlated data points along the columns, evaluated at different points in time along the rows. Given that $X$ consists of correlated data points, the variation in $X$ can be decomposed into a small number $r$ of orthogonal principal components with $r < n$, resulting in a reduction of the dimension of the problem with only a small loss in information. The principal components from standard PCA are linear combinations (along the rows) of the original data set. If it is suspected that the data set contains non-linearities, a common procedure is to "linearise" the data set using suitable transformations prior to analysis. This approach has the advantage that it retains the simplicity of the underlying principal component analysis (PCA)

**Table 7.** Mapping functions employed for non-linear principal component analysis

| $g_1(X)$ | $g_2(X)$ | $g_3(X)$ | $g_4(X)$ | $g_5(X)$ | $g_6(X)$ | $g_7(X)$ | $g_8(X)$ | $g_9(X)$ | $g_{10}(X)$ |
|---|---|---|---|---|---|---|---|---|---|
| $4X(1\text{-}X)$ | $1\text{-}1.4X_t^2+0.3X_{t-1}$ | $0.25X_{t-1}+\varepsilon$ | $\exp(X)$ | $\sin(X)$ | $\cos(X)$ | $X_{t-1}$ | $X_{t-2}$ | $X_t\text{-}X_{t-1}$ | $X_t\text{-}X_{t-2}$ |

whilst gaining the ability to cope with non-linear data. To do this we construct a modified data set $X_{NL}$ from the original data set $X$:

$$X_{NL} = G(X), \tag{4}$$

where $G$ is a function consisting of $n$ individual mapping functions from linear to non-linear space:

$$G = w_1 g_1(X) + w_2 g_2(X) + \cdots + w_n g_n(X), \tag{5}$$

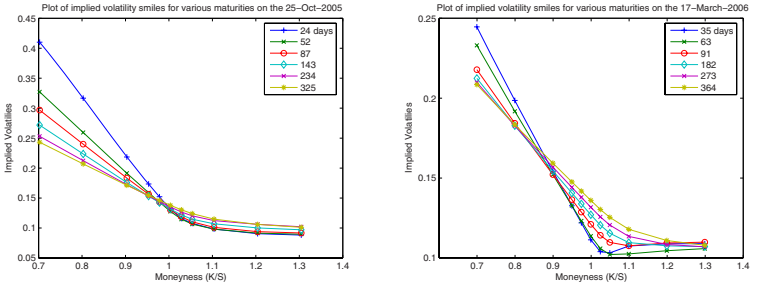
and where $g_i(X)$ is an individual non-linear mapping function of $X$ and $w_i$ is the weight on the function $g_i$. There are an infinite number of mapping functions $g_i(X)$ to choose from and in this paper we consider a small number of mapping functions we think are important given the domain knowledge of the problem under consideration. There are a total of ten functions chosen in this study, including time-series models, and they are given in Table 7.

The previous evidence for PCA applied to implied volatility smiles ([10,12,7]) suggests that changes in the implied volatility smile are driven predominantly by three factors. The first factor is a level factor which controls the overall height of the implied volatility smile. The second and third factors are slope and curvature factors across the strike price axis. However options and the implied volatilities associated with options are multi-dimensional non-linear instruments and standard PCA may neglect some of non-linear subleties inherent in option implied volatilities. This is the reason NLPCA is applied to the IVS in this paper.

## 5 Results

### 5.1 Data

The data used in this study are option implied volatilities across 11 different strikes and a number of different maturities on the FTSE 100 index. The data consists of end-of-day settlement option implied volatilities from the 26th of March 2004 till the 17th of March 2006 consisting of 500 trading days. FTSE 100 index options are European style options and the underlying asset is the FTSE 100 performance index. To price options on this index one must adjust the index by extracting the present value of future cash dividend payments before each options expiration date. The annualised dividend yield of the FTSE 100 index was downloaded from Datastream. The one-month LIBOR rate was used as the risk-free rate where the LIBOR rate was converted into a continuously compounded rate. The forward price used in the option calculations is then simply $F_t = S_0 e^{(r-q)t}$ where $S_0$ is the current index price level, $F_t$ is the price for the forward contract maturing at time $t$, $r$ is the continuously compounded risk-free rate and $q$ is the continuously compounded dividend yield.




(a) 05-Oct-2005 (b) 17-Mar-2006

**Fig. 1.** Implied Volatility Smiles on two different dates

We have interpolated implied volatilities on a fixed grid of moneyness and maturity for all the days in the data sample. For each day $t$ in the sample we define the implied volatility smile at a fixed moneyness $n_m$ and maturity $\tau_j$ by

$$IVS(t) = \{I_t(1, \tau_j), \ldots, I_t(n_m, \tau_j)\}\,.$$

We then stack these implied volatility smiles over time to form the data matrix $X = \{IVS(1), \ldots, IVS(500)\}'$. Non-linear principal component analysis (NLPCA) is conducted on the implied volatility smile matrix for maturities ranging from 2 to 6 months.

### 5.2 Result Analysis

The first three principal components from linear PCA explain up to approximately 96% of the variation in the level of the implied volatility smile, depending on the maturity of the IVS considered. As 96% in PCA analysis may be overfitting, we would rather target the first principal component than the first three components and this why the objective function in the NLPCA was chosen to be proportion of variation explained by the first principal component.

The analysis of the eigenfactors from standard PCA for the implied volatility smiles of each maturity shows that the first factor has a positive effect on all implied volatilities. This eigenfactor can be interpreted as a level or a volatility factor. An increase in this factor causes the whole IVS to increase and causes all options to become more expensive since options are increasing functions of volatility. The second factor has a negative effect for implied volatilities with $K < S$, e.g. out-of-the-money puts, and a positive effect for implied volatilities with $K > S$, e.g. out-of-the-money calls. This factor can be interpreted as a skew factor and increase in this factor causes out-of-the money calls to become more expensive relative to out-of-the-money puts. The third factor has a positive effect for implied volatilities with $K < S$ and $K > S$ e.g. out-of-the-money calls and puts, and a negative effect for implied volatilities that are close to the money with $K \approx S$. This factor can be interpreted as a curvature factor an an increase in this factor causes out-of-the money calls and puts to become more expensive relative to near-the-money calls and puts.

**Table 8.** Results of non-linear PCA. The proportion explained by the first principal component (PC) from the last generation are averaged over 30 runs and compared with the parameter values from 30 runs of a Matlab optimiser.

| Maturity | Linear PCA(%) | Non-linear PCA-QIEA(%) | Non-linear PCA-GA(%) |
|---|---|---|---|
| 2 months | 64.15 | 82.19 | 81.35 |
| 3 months | 69.57 | 82.67 | 82.01 |
| 4 months | 72.90 | 83.94 | 83.21 |
| 5 months | 77.01 | 84.01 | 83.93 |
| 6 months | 80.27 | 83.23 | 82.39 |

In our NLPCA-QIEA analysis, the weights on the mapping functions are optimised by using a quantum-inspired evolutionary algorithm to maximise the objective function which is the proportion of variation in the data explained by the first principal component. The weights are also optimised using the GA Matlab toolbox developed by Andrew Chipperfield. Fig. 2 depicts the evolution of the objective function versus the generation number. The parameter settings in the QIEA are given in Table 2. NLPCA is more efficient than linear PCA especially for the options with shorter times-to-maturity. For example, for the 2 month IVS the 1st principal component from NLPCA explains approximately 82% of the variation of the data versus only 64% for standard PCA. However the outperformance of NLPCA is to be expected given the extra degrees of freedom involved since it uses four non-linear functions that first operate on the data before PCA is applied. It is interesting to note that for the two month IVS the first component from NLPCA with ten non-linear functions explains 82% of the variation whilst the first *three* components from linear PCA explain up to 96% of the variation in the data. Although 96% is a higher level of explanatory power this is more than likely overfitting historical data at the expense of poor out-of-sample performance. If we forecast the evolution of the IVS out-of-sample using the techniques in this paper, a parsimonious procedure would be to include a more general set of time series models in the set of non-linear functions and use these to forecast the first factor from NLPCA and reconstruct future IVS's from the weights derived from historical analysis. This would be more parsimonious than fitting a separate time series model to three linear principal components and then reconstructing the future IVS as would have to be done in linear PCA. Thus, at least for shorter term options, the NLPCA method can explain 82% of the variation in the data with one linear combination of non-linear functions of the data versus approximately 64% for linear PCA. Thus rather than increasing the number of principal components in the analysis we have shown that another route is to use non-linear principal components to achieve a statistical significant increase in explanatory power.

It is interesting to note the weights on various functions that were derived in the NLPCA. The $f_2$ (Hénon function) captures a curvature effect (mentioned earlier) due to the squaring of the data, and a time series effect due to the dependence on past values. The weight on this function is close to one implying that the function depending on the curvature of the current IVS and the past

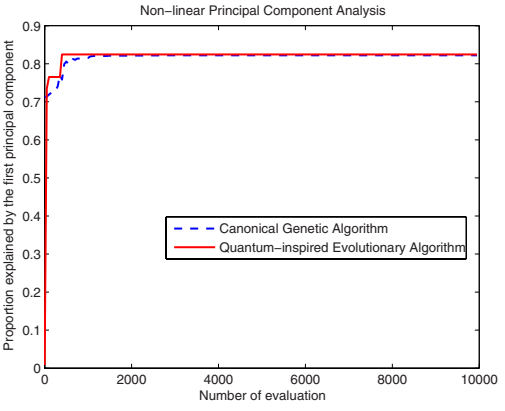



**Fig. 2.** Global search

**Table 9.** Weights on mapping functions. These weights on mapping functions from the last generation are averaged over 30 runs.

| Maturity | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ | $f_8$ | $f_9$ | $f_{10}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 2 months | 0.124 | 0.925 | 0.328 | 0.024 | 0.036 | 0.897 | 0.106 | 0.152 | 0.194 | 0.375 |
| 3 months | 0.121 | 0.901 | 0.371 | 0.016 | 0.047 | 0.923 | 0.097 | 0.133 | 0.189 | 0.342 |
| 4 months | 0.151 | 0.917 | 0.368 | 0.020 | 0.068 | 0.925 | 0.072 | 0.067 | 0.169 | 0.274 |
| 5 months | 0.141 | 0.917 | 0.324 | 0.016 | 0.059 | 0.918 | 0.095 | 0.102 | 0.185 | 0.293 |
| 6 months | 0.137 | 0.882 | 0.363 | 0.019 | 0.066 | 0.903 | 0.074 | 0.073 | 0.168 | 0.247 |

level of the IVS is very important for explaining the variation in the IVS over time. The $f_3$ (auto regressive function) is capturing serial correlation in the daily movements of the IVS (something that cannot be done under linear PCA). Thus there is positive serial correlation in the data and this represents a possible trading strategy.

The weight on $cos(X)$ is approximately 0.9, which means this function contribute much during the mapping process. Also, both QIEA and GA can find the global optima. As they are sensitive to the parameters, i.e. crossover and mutation rate in GA, enlargement, shrinkage, and resize factor in QIEA, further analysis would need to be conducted for sensitivity test, other transformation functions and methods, such as Fourier transformation.

## 6 Conclusions

The results of benchmark tests suggest the potential of QIEA for application to high-dimensional problems. This is particularly interesting for financial applications which often require optimization in complex and high-dimensional environments.

A non-linear principal component analysis was conducted on the implied volatility smile derived from FTSE 100 stock index options. The weights on these non-linear functions were optimised using a QIEA. It has potential to be a highly non-linear non-convex optimisation problem due to the fact that the options data analysed are highly non-linear and method used to describe the variation in the options data is a non-linear method. Thus it was thought that this was a reasonable problem to test out the QIEA. It was shown, at least for shorter term options, that the NLPCA method can explain 82% of the variation in the data with one non-linear principal component versus approximately 64% for one linear principal component in linear PCA. Thus the non-linear functions used in the NLPCA captured some of the higher order non-linear factors that affect the data and increased the explanatory power of the method.

Future work will consist of follow-up benchmark studies on the QIEA to examine both its scalability and its potential utility for optimization in dynamic environments. In the context of the examination of the implied volatility smile, future work consists of expanding the number of non-linear functions being considered with a focus on including a larger number of time series models. This would be very useful in predicting the IVS out-of-sample and in constructing options trading strategies. Future work could also look at multi-objective NLPCA where the proportion of variation explained by the first factor is maximised followed by the proportion of variation explained by the second factor, etc. Also it would be useful to relax the restriction on the parameters of the non-linear functions used in NLPCA and allow the QIEA to find optimal values for these parameters. All of these extensions will result in very high-dimensional optimisation problems where the use of evolutionary algorithms such as the QIEA may be essential.

## References

1. Brabazon, A., O'Neill, M.: Biologically-inspired Algorithms for Financial Modelling. Springer, Berlin (2006)
2. da Cruz, A., Barbosa, C., Pacheco, M., Vellasco, M.: Quantum-Inspired Evolutionary Algorithms and Its Application to Numerical Optimization Problems. In: Pal, N.R., Kasabov, N., Mudi, R.K., Pal, S., Parui, S.K. (eds.) ICONIP 2004. LNCS, vol. 3316, pp. 212–217. Springer, Heidelberg (2004)
3. da Cruz, A., Vellasco, M., Pacheco, M.: Quantum-inspired evolutionary algorithm for numerical optimization. In: Proceedings of the 2006 IEEE Congress on Evolutionary Computation (CEC 2006), Vancouver, 16-21 July, pp. 9180–9187. IEEE Press, Los Alamitos (2006)
4. Fan, K., Brabazon, A., O'Sullivan, C., O'Neill, M.: Quantum-Inspired Evolutionary Algorithms for Calibration of the VG Option Pricing Model. In: Giacobini, M. (ed.) EvoWorkshops 2007. LNCS, vol. 4448, pp. 189–198. Springer, Heidelberg (2007)
5. Fan, K., Brabazon, A., O'Sullivan, C., O'Neill, M.: Option Pricing Model Calibration using a Real-valued Quantum-inspired Evolutionary Algorithm. In: GECCO 2007, pp. 1983–1990. ACM Press, New York (2007)

6. Fan, K., O'Sullivan, C., Brabazon, A., O'Neill, M.: Testing a Quantum-inspired Evolutionary Algorithm by Applying It to Non-linear Principal Component Analysis of the Implied Volatility Smile. In: Natural Computing in Computational Finance, Springer, Heidelberg (in press, 2008)
7. Fengler, M., Härdle, W., Schmidt, P.: Common factors governing VDAX movements and the maximum loss. Jounal of Financial Markets and Portfolio Management 1, 16–19 (2002)
8. Han, K.-H., Kim, J.-H.: Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. IEEE Transactions on Evolutionary Computation 6(6), 580–593 (2002)
9. Han, K.-H., Kim, J.-H.: On setting the parameters of quantum-inspired evolutionary algorithm for practical applications. In: Proceedings of IEEE Congress on Evolutionary Computing (CEC 2003), August 8–December 12, 2003, pp. 178–184. IEEE Press, Los Alamitos (2003)
10. Heynen, R., Kemma, K., Vorst, T.: Analysis of the term structure of implied volatilities. Journal of Financial and Quantitative Analysis 29, 31–56 (1994)
11. Narayanan, A., Moore, M.: Quantum-inspired genetic algorithms. In: Proceedings of IEEE International Conference on Evolutionary Computation, May 1996, pp. 61–66. IEEE Press, Los Alamitos (1996)
12. Skiadopoulos, G., Hodges, S., Clewlow, L.: The Dynamics of the S&P 500 Implied Volatility Surface. Review of Derivatives Research 3, 263–282 (1999)
13. Yang, S., Wang, M., Jiao, L.: A novel quantum evolutionary algorithm and its application. In: Proceedings of IEEE Congress on Evolutionary Computation 2004 (CEC 2004), June 19-23, 2004, pp. 820–826. IEEE Press, Los Alamitos (2004)