

A Versatile Quantum-inspired Evolutionary Algorithm

Michaël Defoin Platel[†], Stefan Schliebs[†] and Nikola Kasabov[†]

Abstract—This study points out some weaknesses of existing Quantum-Inspired Evolutionary Algorithms (QEA) and explains in particular how hitchhiking phenomena can slow down the discovery of optimal solutions and encourage premature convergence. A new algorithm, called Versatile Quantum-inspired Evolutionary Algorithm (vQEA), is proposed. With vQEA, the attractors moving the population through the search space are replaced at every generation without considering their fitness. The new algorithm is much more reactive. It always adapts the search toward the last promising solution found thus leading to a smoother and more efficient exploration. In this paper, vQEA is tested and compared to a Classical Genetic Algorithm CGA and to a QEA on several benchmark problems. Experiments have shown that vQEA performs better than both CGA and QEA in terms of speed and accuracy. It is a highly scalable algorithm as well. Finally, the properties of the vQEA are discussed and compared to Estimation of Distribution Algorithms (EDA).

I. INTRODUCTION

Quantum-Inspired Evolutionary Algorithms (QEA) apply Quantum Computing Principles to enhance classical Evolutionary Algorithms (EA). In the last ten years QEA received a lot of attention and have already demonstrated their superiority compared to classical EA for solving complex benchmark problems such as combinatorial [7], numerical [9], [2] and multiobjective optimization [17], as well as real world problems namely disk allocation method [14], face detection [12], rigid image registration [3], training of multi layer perceptron [18], signal processing [15] and clustering of gene expression data [21]. However QEA are still poorly understood and their integration into the theory of Evolutionary Computation is missing. The main reason for that is probably because of a lack in a unified definition of QEA.

We think that the most illustrative example of QEA is the algorithm firstly proposed by Han and Kim in [7] where some major principles of Quantum Computing are used such as, the quantum and collapsed bit, the linear superposition of states and the quantum rotation gate. This algorithm has been studied several times in terms of both experimental and theoretical behavior, and tested on ideal cases, on classical optimization benchmarks but also on real world problems, [7], [6], [14], [9], [12] and [10].

Nevertheless, we think that some specific characteristics of QEA have not received enough attention yet. In section II-A we recall briefly some basic quantum principles inspiring QEA, then a revisited description of its features is provided in section II-B. Some new tools for exploring the dynamics

of QEA are proposed and a clear trend in promoting hitchhiking is demonstrated. In section II-B we introduce a novel algorithm called Versatile Quantum-inspired Evolutionary Algorithm (vQEA) which intends to remove elitism from the evolutionary process. With vQEA the information about the search space collected during evolution is not kept at the individual level but continuously renewed and shared among the whole population. In section IV, vQEA is tested on different benchmark problems and compared to the classical versions of EA and QEA. Finally in section V, the role of elitism is discussed in the light of Estimation of Distribution Algorithms (EDA).

II. QUANTUM-INSPIRED EVOLUTIONARY ALGORITHM

Quantum-Inspired Evolutionary Algorithms (QEA) apply Quantum Computing Principles to enhance classical Evolutionary Algorithms (EA).

A. Quantum Computing Principles

The smallest information unit in today's digital computers is one bit being either in the state "1" or "0" at any given time. The corresponding analogue on a quantum computer is represented by a quantum bit or Qbit [11]. Similar to classical bits, a Qbit may be in "1"-state or "0"-state but additionally also in any superposition of both states. A Qbit state $|\Psi\rangle$ can be defined as

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad (1)$$

where α and β are complex numbers defining probabilities at which the corresponding state is likely to appear when a Qbit is *collapsed*, i.e. read or measured. In another word, the probability of a Qbit to collapse to state "0" and "1" is $|\alpha|^2$ and $|\beta|^2$ respectively¹. In a more geometrical aspect, an angle θ is defined such that $\cos(\theta) = |\alpha|$ and $\sin(\theta) = |\beta|$.

In order to modify the probabilities α and β , *quantum gates* can be applied. We note that several quantum gates have been proposed such as (controlled) NOT-gate, rotation gate and Hadamard gate, see [11] for details.

B. Description of the QEA

In this section we propose a revisited description of the QEA, originally published in [7], see [6] for a comprehensive definition. QEA is a generational population-based search method which behavior can be decomposed in three different and interacting levels, see Figure 1.

[†]Knowledge Engineering and Research Institute (KEDRI) www.kedri.info Auckland University of Technology (AUT) 581-585 Great South Road, Penrose Auckland, New Zealand

¹Normalization of the states to unity guarantees $|\alpha|^2 + |\beta|^2 = 1$ at any time.

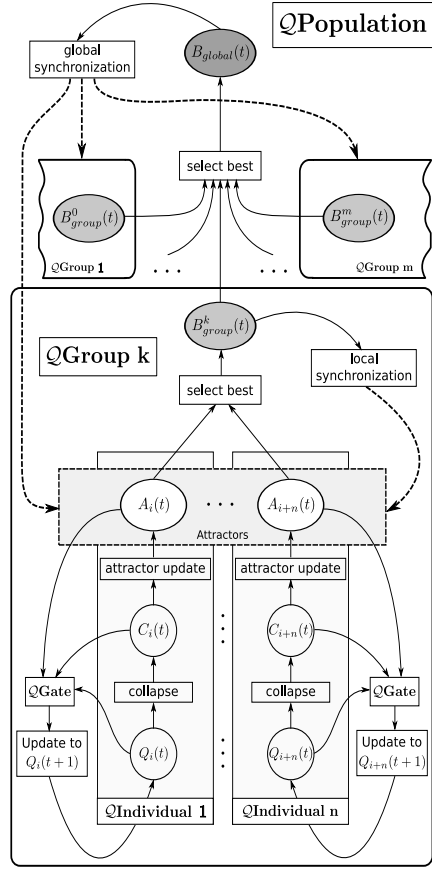


Fig. 1. Description of the QEA with three levels

1) *Quantum individuals:* The lowest level corresponds to *quantum individuals*. A Q individual i at generation t is noted $Q_i(t)$ and corresponds to a string of λ concatenated Q bits:

$$Q_i = Q_i^1 Q_i^2 \dots Q_i^\lambda = \begin{bmatrix} \alpha_i^1 & \alpha_i^2 & \dots & \alpha_i^\lambda \\ \beta_i^1 & \beta_i^2 & \dots & \beta_i^\lambda \end{bmatrix} \quad (2)$$

Each Q individual has to be viewed as a distribution of bit strings of length λ . Even if a Q individual is unchanged, its fitness is reevaluated every generation according to a realization of the distribution. For that purpose, each Q_i is first measured to form a binary individual C_i in a collapsed state² and then the fitness evaluation takes place. In the sense of classical EA, Q_i is the genotype while C_i is the phenotype of a given individual.

To each individual Q_i is also attached a binary string A_i acting as an attractor for Q_i . Indeed, every generation C_i and A_i are compared in terms of both fitness and bit values. If A_i is better than C_i and if their bit values differ, a quantum

²The way a Q_i collapses is described in [7]

gate operator is applied on the corresponding Q bits of Q_i . Thus the distribution Q_i is slightly moved toward a given solution A_i of the search space. We note that if C_i is better than A_i the individual attractor is updated.

In classical EA variation operators like crossover or mutation operations are used to explore the search space. The quantum analogue for these operators is called a quantum gate. In this study, the rotation gate is used to modify the Q bits in a solution. The j^{th} Q bit at time t of Q_i is updated as follows:

$$\begin{bmatrix} \alpha_i^j(t+1) \\ \beta_i^j(t+1) \end{bmatrix} = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix} \begin{bmatrix} \alpha_i^j(t) \\ \beta_i^j(t) \end{bmatrix} \quad (3)$$

where the constant $\Delta\theta$ is a rotation angle designed in compliance with the application problem [8]. We note that the sign of $\Delta\theta$ determines the sense of rotation (clockwise for negative values).

2) *Quantum groups:* The second level corresponds to *quantum groups*. The population is divided into m Q groups each containing n Q individuals having the ability of synchronizing their attractors. For that purpose, the best attractor (in terms of fitness) of a group, noted B_{group} , is stored at every generation and is periodically distributed to the group attractors. This phase of local synchronization is controlled by the parameter S_{local} .

3) *Quantum population:* The set of all $n \times m$ Q individuals forms the *quantum population* and defines the topmost level of QEA. As for Q groups, the individuals of a Q population can synchronize their attractors. For that purpose, the best attractor (in terms of fitness) among all Q groups, noted B_{global} , is stored every generation and is periodically distributed to the group attractors. This phase of global synchronization is controlled by the parameter S_{global} . We note that in the initial population all the Q bits are fixed with $|\alpha|^2 = |\beta|^2 = 1/2$ so that the two states "0" and "1" are equiprobable in collapsed individuals.

C. QEA on the OneMax problem

The OneMax Problem consists of maximizing the number of ones of a bit string and the global optimum is noted 1^λ . In this section the behavior of QEA on the OneMax problem is studied for $\lambda=100$. For that purpose new tools for monitoring the dynamics of both Q individuals and Q bits are used. The setting of the evolutionary parameters is similar to the settings proposed in [7], with a population of 10 individuals, 5 groups, $\Delta\theta = \pm 0.01\pi$, $S_{local}=1$ and $S_{global}=100$. The Figure 2 presents the typical evolution of the 100 Q bits of the Q individual Q_6 on the OneMax problem. Each point $Q_6^j(t)$ corresponds to a given Q bit j and a given generation t . The color (gray scale) indicates the value of the corresponding $|\beta|^2$, from black for $|\beta|^2 = 0$ to white for $|\beta|^2 = 1.0$. Thus, a Q individual with all Q bits such that $|\beta|^2 \simeq 1.0$ is likely to collapsed into the global optimum 1^λ . We see that the evolutionary process starts by construction with initial $|\beta|^2$ values equal to $1/2$. Most of the Q bits evolve toward the optimum as the color changes to white. Nevertheless

we can see clearly that some Q bits are rotated toward the wrong direction as some very dark points appear. For the vast majority of them, they finally move toward the expected value with $|\beta|^2$ close to 1, but one of them Q_6^{33} has converged with $|\beta|^2$ close to 0. For this run QEA was not able to find the global optimum in 500 generations.

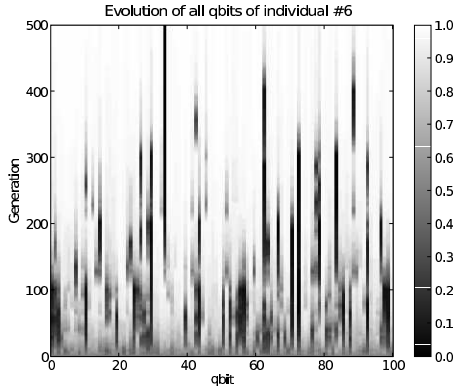


Fig. 2. Typical evolution of a Q individual (value of $|\beta|^2$) with QEA on the OneMax problem

To understand this unappropriated behavior of Q_6^{33} , we have plotted its evolution, *i.e.* values of $|\alpha|$ (dotted line) and $|\beta|$ (solid line), as well as the states of the corresponding collapsed bit C_6^{33} and attractor bit A_6^{33} , *cf.* Figure 3. We see that the $|\beta|$ value is converging toward 0 as of the first generations and as a consequence the state of the collapsed bit is most of the time 0. We note also that the state of the attractor bit demonstrates few variations and is also most of the time 0, except for two very short periods before generation 100. An attractor is always chosen according to its fitness. So the attractor A_6 is always better than the collapsed individual C_6 even if the value of its 33th Q bit is not well adapted.

D. Hitchhiking and the irreversible choice

A quantum individual Q_i explores a given region of the search space. If a good solution is found in this region, this one is chosen as an attractor and the exploration will concentrate on this new area. In the general case, two ways exist for an attractor A_i to be updated, either when a better collapsed individual C_i is found, or when a synchronization phase occurs. At the individual level, when a new attractor A_i is chosen in the search space, the corresponding Q_i will be slightly moved toward this point until a better C_i is found. But what if not better C_i is found during this move? Then

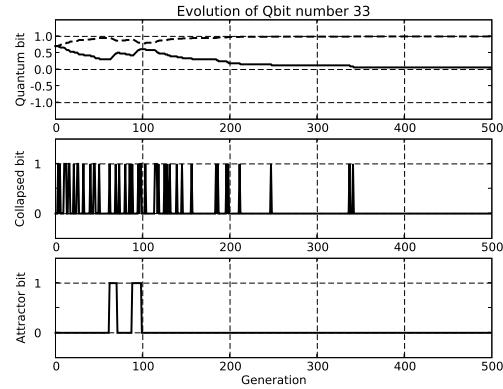


Fig. 3. Typical evolution of a Quantum bit, Collapsed bit and Attractor bit with QEA on the OneMax problem

the algorithm is trapped and converges prematurely to this point. The only opportunity for an individual to escape from this attractor is that a synchronization phase replaces it with a better attractor produced elsewhere. Otherwise it is possible that the choice of a very good but non optimal attractor is irreversible.

We think that the weakness of QEA described here is similar to what happens in Classical Genetic Algorithms (CGA) with the so called *hitchhiking* phenomenon firstly described as a serious bottleneck for CGA in [4]. Hitchhiking corresponds to the increase in frequency of a “bad” allele at a given locus in the population due to the presence of nearby highly fit alleles on the same chromosomes [5]. As a consequence, the eventual better alleles at the same locus (as the hitchhiking allele) tend to disappear in the population and there is no way for the evolutionary process to retrieve them. In CGA, random mutation and uniform crossover are two known remedies against hitchhiking.

III. VERSATILE QUANTUM-INSPIRED EVOLUTIONARY ALGORITHM

In this section we present an improved version of QEA, called the Versatile Quantum-inspired Evolutionary Algorithm (vQEA) avoiding the weaknesses reported above.

A. Description of vQEA

In order to prevent both the case of irreversible choice and the hitchhiking phenomenon, the strategy for updating attractors is modified. We introduce a new parameter controlling this strategy based on *elitism*. In the classical QEA, the update procedure (called *attractor update* in Figure 1) applies elitism such that an attractor A_i is replaced by C_i only if C_i is better. With vQEA this parameter is simply switched off. Therefore the attractors are replaced at every generation without considering their fitness and so they demonstrate a very high volatility. Moreover to ensure the convergence of vQEA, the global synchronization is also performed every

generation in such way that all the attractors are identical and at generation $t+1$ corresponds to the best solution found at generation t .

We note that with such a setting, the group size n and local synchronization parameters S_{local} do not affect the algorithm anymore. With vQEA the information about the search space collected during evolution is not kept at the individual level but continuously renewed and shared among the whole population³. Nevertheless we think that the concept of group, which is similar to demes in classical EA, is interesting and we do not intend to removed it definitely. In this study, we avoid the tuning of n and S_{local} and concentrate on the effects of removing elitism from QEA. Thus the simplified sequential procedure of vQEA is detailed in Algorithm 1. The sets of all the quantum individuals, collapsed individuals and attractors at generation t are noted $Q(t)$, $C(t)$ and $A(t)$ respectively.

Algorithm 1 The Versatile Quantum-inspired EA (vQEA)

- 1: $t \leftarrow 0$
 - 2: initialize $Q(t)$ and $A(t)$
 - 3: **while** not termination condition **do**
 - 4: make $C(t)$ by observing the states of $Q(t)$
 - 5: evaluate $C(t)$
 - 6: update $Q(t)$ according to $C(t)$ and $A(t)$ using $\mathcal{Q}Gate$
 - 7: store the best of $C(t)$ into $B_{global}(t)$
 - 8: synchronize $A(t)$ with $B_{global}(t)$
 - 9: $t \leftarrow t + 1$
 - 10: **end while**
-

B. vQEA on the OneMax problem

The behavior of vQEA on the OneMax problem is studied for $\lambda = 100$. The setting of the evolutionary parameters is kept almost unchanged to allow fair comparison with QEA, with a population of 10 individuals, $\Delta\theta = \pm 0.01\pi$, of course no elitism and global synchronization every generation (local synchronization and number of group being meaningless).

We have plotted Figure 4 the evolution of two illustrative \mathcal{Q} bits for QEA (dashed line) and vQEA (solid line) on the OneMax problem. Actually the value of $\theta(t)$ is reported in the polar coordinates system, the radius is given by t and the angle corresponds to θ such that $\cos(\theta) = |\alpha|$ and $\sin(\theta) = |\beta|$. For both algorithms a successful run is presented since for both cases the angle θ finally reached an expected value close to 90 degree, i.e. β close to 1.0. Nevertheless it is clear that QEA and vQEA display a very different behavior. With QEA strong decisions are made and when a rotation sense is chosen this one is kept during several generations. In fact this constancy is related to the strategy adopted for updating the attractors that is based on elitism. Conversely for vQEA, the trajectory of $\theta(t)$ is much more unsettled and during the first 300 generations, a high number of variations is reported.

³It is worth noticing that an extra long term memory mechanism has been added to store the best collapsed individual found ever, but this mechanism does not influence the algorithm

Nevertheless, the overall evolution is much smoother than with the classical QEA.

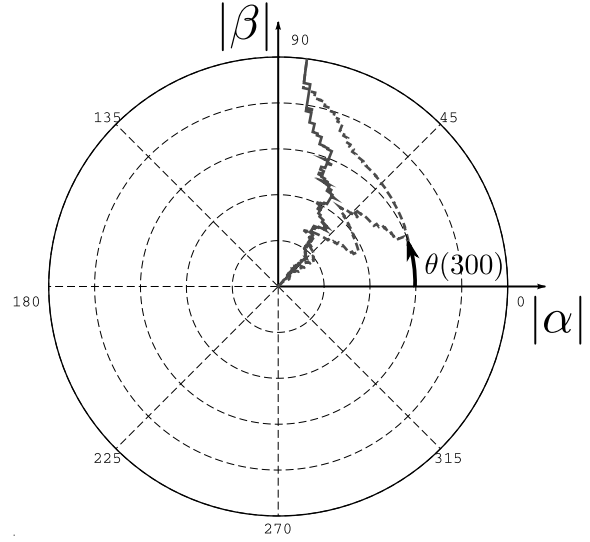


Fig. 4. Typical evolution of a Quantum bit (value of $\theta(t)$) for QEA (dashed line) and vQEA (solid line) on the OneMax problem

To illustrate this situation, we have also computed for both algorithms the average total number of different attractors used per individual during one run of 500 generations on the OneMax problem. We found 25.5 for QEA and more than 372 for vQEA, meaning that the “life duration” of an attractor is around 19.6 generations for QEA and only 1.34 generation for vQEA .

The Figure 5 presents the typical evolution with vQEA of the 100 \mathcal{Q} bits of the \mathcal{Q} individual Q_5 on the OneMax problem. We can see a phase of more than 100 generations where the \mathcal{Q} individual stay undecided. Then all the \mathcal{Q} bits evolve slowly toward the optimum as the color changes to white. We note that for this run vQEA was able to find the global optimum in 298 generations.

To understand this behavior of \mathcal{Q} bits, we have plotted the evolution of Q_5^{17} , i.e. values of $|\alpha|$ (dotted line) and $|\beta|$ (solid line), as well as the states of the corresponding collapsed bit C_5^{17} and attractor bit A_5^{17} , cf. Figure 6. We see that the $|\beta|$ value is slowly but continuously moving toward 1 which is expected. Meanwhile the attractor bit reports many changes of its state at the early generations then the frequency is decreasing and finally the attractor bit converges to 1.

IV. EXPERIMENTS

In this section, vQEA is tested and compared to a Classical Genetic Algorithm (CGA) and to a QEA on two benchmark problems. For both problems the fitness of the average best solution found over 30 runs is presented. We use a statistical unpaired, two-tailed t -test with 95% confidence to determine if results are significantly different.

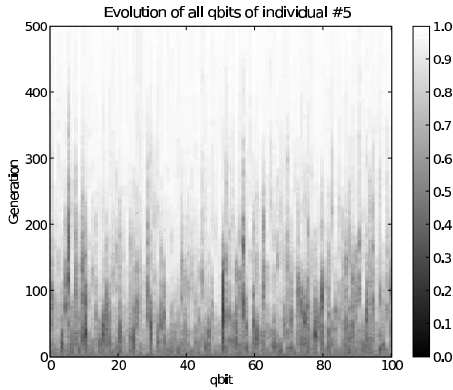


Fig. 5. Typical evolution of a Qindividual (value of $|\beta|^2$) with vQEA on the OneMax problem

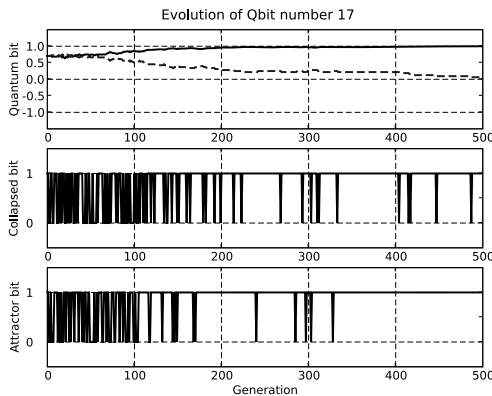


Fig. 6. Typical evolution of a Quantum bit, Collapsed bit and Attractor bit with vQEA on the OneMax problem

A. Optimization of a 01-knapsack problem

This NP-hard problem consists in finding the most valuable subset among N items of different profits and volumes that fit in a knapsack of limited capacity. In [8], QEA was already evaluated on a 01-knapsack problem. For both CGA and QEA, the same settings for the evolutionary parameters are used here. We note that the population size is equal to 100 for the CGA and only 10 for both QEA and vQEA. For vQEA of course the elitism was switched off and S_{global} set to one. The results are reported in Table I for $N=500$ items. Our implementation of CGA and QEA found solutions comparable to [8]. QEA significantly outperforms the CGA

but the best results came with vQEA. The improvement of the average profit obtained when comparing vQEA to QEA is almost equivalent to the improvement obtained when comparing QEA to CGA.

TABLE I
AVERAGE PROFIT OF THE BEST SOLUTION FOUND ON THE 01-KNAPSACK PROBLEM

| $\lambda = N = 500, 1000$ generations | | |
|---------------------------------------|--------------------------|--------------------------|
| CGA | QEA | vQEA |
| 2963.5 ($\sigma=19.7$) | 3013.5 ($\sigma=18.9$) | 3058.0 ($\sigma=15.9$) |

In Figure 7, the evolution of the average best profit is plotted for the three algorithms. We see that during the first 306 generations the CGA reports the best profit then, until the generation 1000, vQEA outperforms both CGA and QEA. The solutions discovered by CGA and vQEA at generation 306 correspond to an average profit of 2932. This value is reached by the QEA only at generation 454.

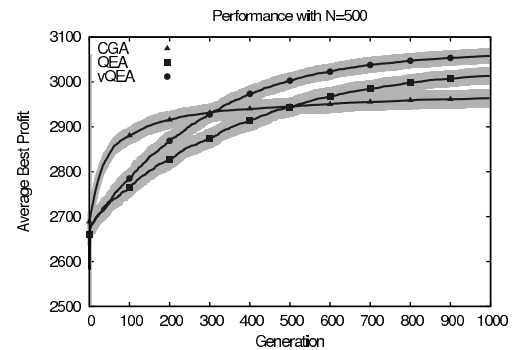


Fig. 7. Average profit of the best solution found on a 01-knapsack problem with $N=500$

B. Optimization of NK -landscapes problems

In [13], Stuart Kauffman developed the NK -landscapes to model systems which performance depends not only on the states of their N components but also on the K interactions between them. They have been used in theoretical biology for example to study gene networks, evolution of proteins or immune systems. The NK -landscapes define also a family of combinatorial optimization problems that are now widely used as benchmarks for EA. According to Weinberger [19], the model affords a “tunably rugged” fitness landscape. The parameter N determines the size of the search space while K controls the number of local optima, from no local optima for $K = 0$ to $\frac{2^N}{N+1}$ for $K = N - 1$. In this study, the K interactions between the N parts of the systems are chosen randomly and the corresponding problem has been proved to be NP-complete for $K \geq 1$ [19]. The performances of the three algorithms are studied for problems of increasing size with $N=256, 512, 1024, 2048$ and 4096 and of increasing

difficulty with K varying from 0 to 8. Each run corresponds to 10,000 generations.

The average fitness of the best solutions found with $K=0$ and 8 are plotted in Figure 8 (error bars corresponds to three times the standard deviation). In part A, for $K=0$ and $N=256$ the problem is very simple and can be easily solved by the three algorithms. While N increases, we see that both CGA and QEA are outperformed by vQEA. Moreover the average fitness of the solutions found with vQEA is almost unaffected by N . In part B for $K=8$ the performances of the three algorithms are not significantly different for $N=256$ and 512 but for higher N the trend reported for $K=0$ is still present. These results demonstrate that vQEA is a highly scalable algorithm even for difficult problems.

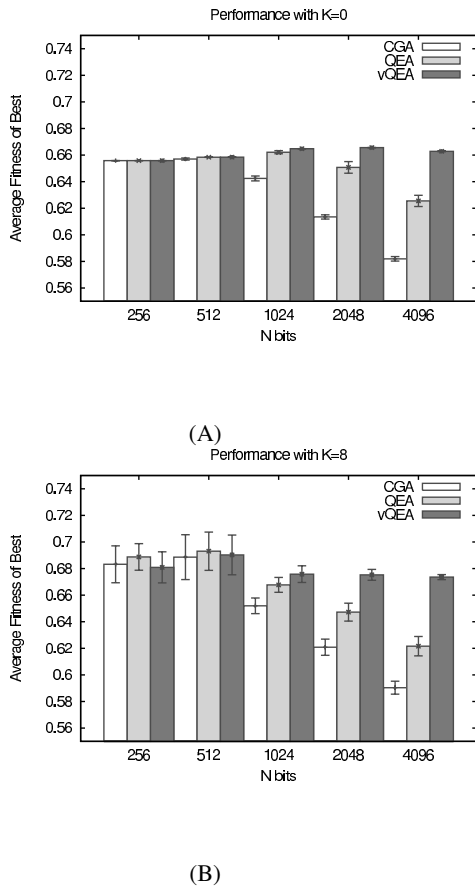


Fig. 8. Average fitness of the best solution found on NK -landscapes with $K=0$ (part a) and 8 (part b)

In Figure 9 we have plotted the relative fitness of the best solutions found for $N=4096$ and for different values of K , *i.e.* the results obtained by the CGA are used as a reference. It is clear that the performance of vQEA compared to CGA are almost independent of the difficulty with vQEA around 14% better than CGA. Conversely for QEA this ratio varies from less than 8% for $K=0$ to nearly 5% for $K=8$. We note

also that the standard deviation reported for vQEA is the smallest, indicating that most of the 30 runs have found nearly the same good solution. For further inter-comparisons,

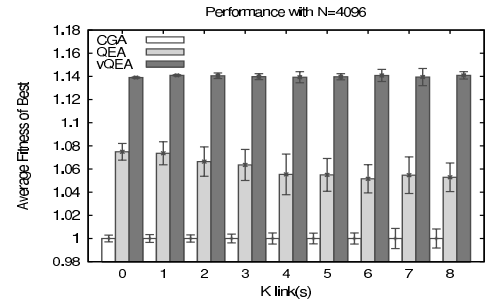


Fig. 9. Relative fitness of the best solution found on NK -landscapes with $N=4096$

the overall numerical results obtained with CGA, QEA and vQEA are reported Table II.

The Figure 10 presents two isofitness clouds. In these clouds, each point of coordinates (gen_A, gen_B) corresponds to the average number of generations, *i.e.* gen_A and gen_B , needed by two different algorithms, respectively A and B , to reach the same fitness value. We introduce this kind of representation to allow practical comparisons of computational resources required by algorithms reporting different best fitness values and different convergence speeds. In our case, the isofitness clouds have been computed from all the experiments on the NK -landscapes reported above. It is worth noticing that the underlying assumption is that the resources needed for computing one generation are equivalent between all the algorithms tested, which is partly false. Indeed, when CGA and QEA (same for vQEA) are compared, the size of the populations are significantly different, respectively 100 and 10 individuals and so a generation is processed faster with QEA or vQEA than with CGA.

In part (A) CGA and QEA are compared. We see that most of the points are located under the line $y = x$ showing that QEA was faster than CGA. The biggest difference in convergence speed is reported for points at the bottom right corner of the figure meaning that it requires 10,000 generations to CGA to discover solutions which fitness was found by QEA around generation 1,000. However we note that for the early generations, *i.e.* before 1000, some points indicate that CGA was the first to reach a given fitness level. After studying the data, we have found that those points correspond to the easiest problems with small values of N (256 and 512) and K equal to 0. The part (B) displays the isofitness cloud obtained for QEA vs vQEA. It is clear that vQEA is almost always faster than QEA whatever the size and the difficulty of the problem. We see also that, after the generation 4,000, the "slope" of the cloud is nearly equal to 0. This means that the QEA needs a very high number (asymptotically an infinite number) of generations to find

solutions as good as the solutions found by vQEA in less than 2,000 generations.

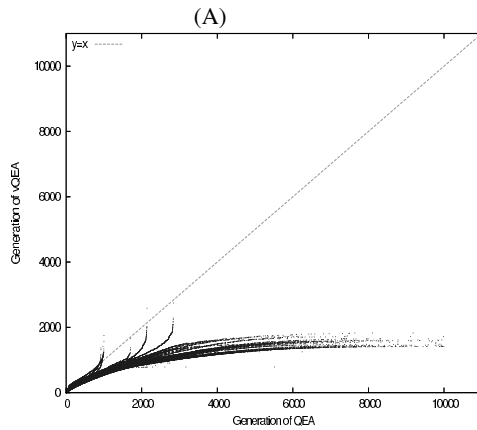
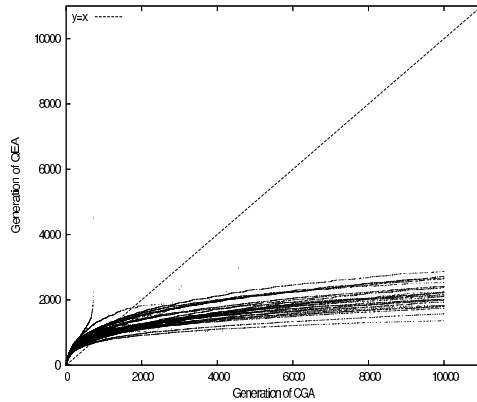


Fig. 10. Isotfitness Clouds: part (a) CGA vs QEA , part (b) QEA vs vQEA

V. DISCUSSION

According to [16], the algorithms that use a probabilistic model of promising solutions to guide further exploration of the search space are called Estimation of Distribution Algorithms (EDA). We think that in QEA the Q individuals act as a probabilistic models and so, as it has already been claimed in [20] and [10], QEA is a new approach belonging to EDA. In this section the role of elitism in EDA is briefly discussed.

In CGA elitism has been introduced as a protection mechanism to counteract the disruptive effects of genetic operators such as the uniform crossover. In some EDA the probabilistic models can undergo perturbations to explore the search space but these perturbations do not have dramatic consequences

TABLE II
AVERAGE PROFIT OF THE BEST SOLUTION FOUND ON THE
 NK -LANDSCAPES PROBLEM AFTER 10,000 GENERATIONS

| K | CGA | QEA | vQEA |
|-----|-------------------------|-------------------------|-------------------------|
| | $N = 256$ | | |
| 0 | 0.655($\sigma=0.000$) | 0.655($\sigma=0.000$) | 0.655($\sigma=0.000$) |
| 1 | 0.677($\sigma=0.000$) | 0.677($\sigma=0.000$) | 0.677($\sigma=0.000$) |
| 2 | 0.679($\sigma=0.003$) | 0.680($\sigma=0.001$) | 0.680($\sigma=0.000$) |
| 3 | 0.682($\sigma=0.010$) | 0.679($\sigma=0.009$) | 0.674($\sigma=0.005$) |
| 4 | 0.690($\sigma=0.007$) | 0.694($\sigma=0.004$) | 0.695($\sigma=0.002$) |
| 5 | 0.689($\sigma=0.007$) | 0.691($\sigma=0.008$) | 0.683($\sigma=0.006$) |
| 6 | 0.686($\sigma=0.009$) | 0.692($\sigma=0.003$) | 0.691($\sigma=0.004$) |
| 7 | 0.690($\sigma=0.011$) | 0.691($\sigma=0.007$) | 0.695($\sigma=0.009$) |
| 8 | 0.683($\sigma=0.009$) | 0.688($\sigma=0.006$) | 0.680($\sigma=0.007$) |
| K | $N = 512$ | | |
| 0 | 0.657($\sigma=0.000$) | 0.658($\sigma=0.000$) | 0.658($\sigma=0.000$) |
| 1 | 0.681($\sigma=0.000$) | 0.682($\sigma=0.000$) | 0.682($\sigma=0.000$) |
| 2 | 0.671($\sigma=0.002$) | 0.673($\sigma=0.001$) | 0.673($\sigma=0.000$) |
| 3 | 0.673($\sigma=0.005$) | 0.676($\sigma=0.003$) | 0.678($\sigma=0.000$) |
| 4 | 0.681($\sigma=0.003$) | 0.683($\sigma=0.000$) | 0.683($\sigma=0.000$) |
| 5 | 0.679($\sigma=0.006$) | 0.684($\sigma=0.001$) | 0.685($\sigma=0.000$) |
| 6 | 0.687($\sigma=0.011$) | 0.692($\sigma=0.006$) | 0.687($\sigma=0.006$) |
| 7 | 0.678($\sigma=0.003$) | 0.680($\sigma=0.003$) | 0.680($\sigma=0.004$) |
| 8 | 0.688($\sigma=0.011$) | 0.693($\sigma=0.009$) | 0.690($\sigma=0.009$) |
| K | $N = 1024$ | | |
| 0 | 0.642($\sigma=0.001$) | 0.662($\sigma=0.000$) | 0.664($\sigma=0.000$) |
| 1 | 0.648($\sigma=0.001$) | 0.665($\sigma=0.002$) | 0.669($\sigma=0.002$) |
| 2 | 0.643($\sigma=0.001$) | 0.660($\sigma=0.001$) | 0.665($\sigma=0.000$) |
| 3 | 0.649($\sigma=0.002$) | 0.667($\sigma=0.002$) | 0.672($\sigma=0.002$) |
| 4 | 0.653($\sigma=0.003$) | 0.673($\sigma=0.003$) | 0.679($\sigma=0.000$) |
| 5 | 0.658($\sigma=0.003$) | 0.675($\sigma=0.002$) | 0.681($\sigma=0.001$) |
| 6 | 0.653($\sigma=0.002$) | 0.667($\sigma=0.003$) | 0.674($\sigma=0.003$) |
| 7 | 0.654($\sigma=0.004$) | 0.670($\sigma=0.003$) | 0.676($\sigma=0.003$) |
| 8 | 0.651($\sigma=0.003$) | 0.667($\sigma=0.003$) | 0.675($\sigma=0.004$) |
| K | $N = 2048$ | | |
| 0 | 0.613($\sigma=0.001$) | 0.650($\sigma=0.002$) | 0.665($\sigma=0.000$) |
| 1 | 0.612($\sigma=0.001$) | 0.645($\sigma=0.004$) | 0.665($\sigma=0.000$) |
| 2 | 0.617($\sigma=0.001$) | 0.649($\sigma=0.004$) | 0.671($\sigma=0.001$) |
| 3 | 0.617($\sigma=0.002$) | 0.650($\sigma=0.004$) | 0.673($\sigma=0.000$) |
| 4 | 0.623($\sigma=0.001$) | 0.655($\sigma=0.004$) | 0.678($\sigma=0.000$) |
| 5 | 0.617($\sigma=0.002$) | 0.647($\sigma=0.004$) | 0.671($\sigma=0.000$) |
| 6 | 0.624($\sigma=0.002$) | 0.653($\sigma=0.005$) | 0.678($\sigma=0.001$) |
| 7 | 0.623($\sigma=0.004$) | 0.653($\sigma=0.004$) | 0.678($\sigma=0.001$) |
| 8 | 0.620($\sigma=0.004$) | 0.647($\sigma=0.004$) | 0.675($\sigma=0.002$) |
| K | $N = 4096$ | | |
| 0 | 0.581($\sigma=0.001$) | 0.625($\sigma=0.002$) | 0.662($\sigma=0.000$) |
| 1 | 0.586($\sigma=0.001$) | 0.629($\sigma=0.003$) | 0.669($\sigma=0.000$) |
| 2 | 0.585($\sigma=0.001$) | 0.623($\sigma=0.004$) | 0.667($\sigma=0.000$) |
| 3 | 0.587($\sigma=0.001$) | 0.624($\sigma=0.005$) | 0.669($\sigma=0.001$) |
| 4 | 0.587($\sigma=0.001$) | 0.619($\sigma=0.006$) | 0.669($\sigma=0.001$) |
| 5 | 0.587($\sigma=0.001$) | 0.619($\sigma=0.005$) | 0.669($\sigma=0.001$) |
| 6 | 0.589($\sigma=0.001$) | 0.619($\sigma=0.004$) | 0.672($\sigma=0.002$) |
| 7 | 0.589($\sigma=0.003$) | 0.621($\sigma=0.006$) | 0.671($\sigma=0.002$) |
| 8 | 0.590($\sigma=0.003$) | 0.621($\sigma=0.004$) | 0.673($\sigma=0.001$) |

and elitism is not necessary. Moreover with some other EDA, the probabilistic models are completely reconstructed every generation and elitism is not used. Nevertheless we found in [1] an interesting counter example where an EDA is presented and better results are reported with elitism. But in that case a uniform crossover is also applied to bit strings.

So we think that as far as no disruptive operators are employed, there is no need for an EDA and so for a quantum inspired algorithm to have recourse to elitism. This is one the

reasons why vQEA performs better than QEA .

VI. CONCLUSIONS

The Quantum-Inspired Evolutionary Algorithm (QEA) introduced in [7] and studied in this paper is elitist. The exploration of the search space is driven by attractors corresponding to the best solution found so far either at the individual, local or global level. If a non-optimal solution is propagated to the global level then this solution starts to attract the entire population. In that case, to avoid being trapped, the algorithm has to discover a better solution before converging to this global attractor.

To prevent the choice of an attractor from being irreversible, the Versatile Quantum-Inspired Evolutionary Algorithm (vQEA) is proposed. With vQEA, the elitism is switched off and the search at time $t + 1$ is driven by the best solution found at time t . Simply removing elitism has strong consequences. With vQEA the information about the search space collected during evolution is not kept at the individual level but continuously renewed and shared among the whole population. In terms of both speed and accuracy vQEA performs better than QEA on different benchmark problems.

The dynamics of QEA and vQEA are very distinct. The short term behavior of QEA is almost always constant because preferential search directions are chosen and followed during several generations. Conversely with vQEA, the short term behavior is much more unsettled and the search directions are reevaluated every generation. Thus the eventual decision errors do not have long term consequences. vQEA is continuously adapting the search to local information while the quantum individuals act as memory buffers to keep track of the search history. This leads to a much more smooth and efficient long term exploration of the search space.

In this study, since all the attractors are synchronized at every generation, the local level with the Qgroups are meaningless. Nevertheless we think that the concept of group, which is similar to demes in classical EA, is very promising and further studies should address the setting of both local and global synchronization.

In future work the relationship between Quantum-Inspired Evolutionary Algorithms and Estimation of Distribution Algorithms should be investigated strongly, some empirical comparisons should be performed and an unified definition should be proposed.

REFERENCES

- [1] C. W. Ahn, K. P. Kim, and R. S. Ramakrishna. A memory-efficient elitist genetic algorithm. In *Parallel Processing and Applied Mathematics, PPAM2003, 5th International Conference, Proceedings*, pages 552–559, 2003.
- [2] A.V. Abs da Cruz, M.B.R Vellasco, and M.A.C. Pacheco. Quantum-inspired evolutionary algorithm for numerical optimization. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress*, pages 2630–2637, July 2006.
- [3] A. Draa, M. Batouche, and H. Talbi. A quantum-inspired differential evolution algorithm for rigid image registration. In *International Conference on Computational Intelligence, ICCI 2004, Istanbul, Turkey, Proceedings*, pages 408–411, 2004.
- [4] S. Forrest and M. Mitchell. Relative building-block fitness and the building-block hypothesis. In *Foundation of Genetic Algorithms 2*, pages 109–126. Morgan Kaufman, 1993.
- [5] D. J. Futuyma. *Evolutionary Biology*. Sinauer, Sunderland, MA, USA, 1998.
- [6] K-H. Han. *Quantum-inspired Evolutionary Algorithm*. PhD thesis, Korea Advanced Institute of Science and Technology (KAIST), 2003.
- [7] K-H. Han and J-H. Kim. Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. *IEEE Transactions on Evolutionary Computation*, 6(6):580–593, December 2002.
- [8] K-H. Han and J-H. Kim. On setting the parameters of quantum-inspired evolutionary algorithm for practical application. In *Evolutionary Computation. CEC '03. The 2003 Congress*, volume 1, pages 178–194, 2003.
- [9] K-H. Han and J-H. Kim. Quantum-inspired evolutionary algorithms with a new termination criterion, h_c gate, and two phase scheme. *IEEE Transactions on Evolutionary Computation*, 8(2):156–169, 2004.
- [10] K-H. Han and J-H. Kim. On the analysis of the quantum-inspired evolutionary algorithm with a single individual. In *IEEE Congress on Evolutionary Computation*, pages 16–21, 2006.
- [11] T. Hey. Quantum computing: an introduction. *Computing & Control Engineering Journal*, 10:105–112, 06 1999.
- [12] J-H. Kim J-S. Jang, K-H. Han. Face detection using quantum-inspired evolutionary algorithm. In IEEE Press, editor, *the 2004 IEEE Congress on Evolutionary Computation*, pages 2100–2106. IEEE Press, June 2004.
- [13] S. A. Kauffman. “The origins of order”. *Self-organization and selection in evolution*. Oxford University Press, New-York, 1993.
- [14] K-H. Kima, J-Y. Hwang, K-H. Han, J-H. Kim, and K-H. Park. A quantum-inspired evolutionary computing algorithm for disk allocation method. *IEICE Transactions on Information and Systems*, E86-D(3):645–649, March 2003.
- [15] F. Liu, S. Li, M. Liang, and L. Hu. Wideband signal doa estimation based on modified quantum genetic algorithm. *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, 89:648–653, 2006.
- [16] H. Mühlenbein and G. Paass. From recombination of genes to the estimation of distributions i. binary parameters. In *Parallel Problem Solving from Nature - PPSN IV, International Conference on Evolutionary Computation.*, pages 178–187, 1996.
- [17] H. Talbi, A. Draa, and M. Batouche. A novel quantum-inspired evaluation algorithm for multi-source affine image registration. *Int. Arab J. Inf. Technol.*, 3(1):9–15, 2006.
- [18] G. K. Venayagamoorthy and G. Singhal. Quantum-inspired evolutionary algorithms and binary particle swarm optimization for training mlp and srn neural networks. *Journal of Computational and Theoretical Nanoscience*, 2(4):561–568(8), December 2005.
- [19] E. D. Weinberger. NP completeness of Kauffman’s NK model, a tuneably rugged fitness landscape. Technical Report 96-02-003, Santa Fe Institute, 1996.
- [20] S. Zhou and Z. Sun. A new approach belonging to EDAs: Quantum-inspired genetic algorithm with only one chromosome. In *ICNC (3)*, pages 141–150, 2005.
- [21] W. Zhou, C. Zhou, Y. Huang, and Y. Wang. *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*, chapter Analysis of Gene Expression Data: Application of Quantum-Inspired Evolutionary Algorithm to Minimum Sum-of-Squares Clustering, pages 383–391. Springer Berlin / Heidelberg, 2005.