Quantum-Inspired Evolutionary Algorithm: A Multimodel EDA

Michaël Defoin Platel, Stefan Schliebs, and Nikola Kasabov, *Senior Member, IEEE*

***Abstract*—The quantum-inspired evolutionary algorithm (QEA) applies several quantum computing principles to solve optimization problems. In QEA, a population of probabilistic models of promising solutions is used to guide further exploration of the search space. This paper clearly establishes that QEA is an original algorithm that belongs to the class of estimation of distribution algorithms (EDAs), while the common points and specifics of QEA compared to other EDAs are highlighted. The behavior of a versatile QEA relatively to three classical EDAs is extensively studied and comparatively good results are reported in terms of loss of diversity, scalability, solution quality, and robustness to fitness noise. To better understand QEA, two main advantages of the multimodel approach are analyzed in details. First, it is shown that QEA can dynamically adapt the learning speed leading to a smooth and robust convergence behavior. Second, we demonstrate that QEA manipulates more complex distributions of solutions than with a single model approach leading to more efficient optimization of problems with interacting variables.**

***Index Terms*—Coarse grained algorithm, optimization, probabilistic models, quantum computing.**

I. Introduction

NUMEROUS natural and physical real world processes have recently inspired researchers in various domains of artificial intelligence, such as neurocomputing, artificial evolution, ant colony optimization, or simulated annealing, to name a few. The use of metaphoric comparisons is a clear trend of nowadays, especially for search and optimization algorithms. Nevertheless, the metaphor cannot last too long without a strong theoretical justification.

Quantum physics and quantum computing principles have also been widely seen as a source of inspiration, for example in neural networks [1], genetic algorithms [2], differential evolution [3], artificial immune systems [4], and particle swarm optimization [5]. In the field of evolutionary computation the introduction of the quantum-inspired evolutionary algorithms (QEA) by Han and Kim might be the most successful application of the quantum metaphor [6]–[8]. It has been earlier alluded that QEA is related to estimation of distribution algorithms (EDAs) [9], [10]. The first aim of this paper is to integrate in a more systematical way QEA into the class of EDAs as an original algorithm.

Manuscript received September 28, 2007; revised March 28, 2008. First published December 09, 2008; current version published December 01, 2009. This work was supported by the Knowledge and Discovery Research Institute (KEDRI).

The authors are with the Knowledge Engineering and Research Institute (KEDRI), Auckland University of Technology (AUT), Auckland 1010, New Zealand (e-mail: see www.kedri.info).

Digital Object Identifier 10.1109/TEVC.2008.2003010

EDAs have shown their ability to avoid the disruptive effects of genetic operators in evolutionary algorithms (EA), namely crossover and mutation, by iteratively evolving a probabilistic model to explore the search space. Three different classes of EDAs have been proposed, which categorizes these algorithms according to the modeling of interaction between variables of optimization problems [11], see also [12] for an overview of proposed EDAs for each class. Early EDAs assume independent relationship between parameters for a problem and thus the probability distribution of solutions can be factored as a product of independent univariate probabilities. This class of EDAs includes the well-known probabilistic incremental learning (PBIL) [13], the compact genetic algorithm (cGA) [14], and the univariate marginal distribution algorithm (UMDA) [15], to name a few. Recent developments in the field of EDAs take into account possible interactions between variables. Modeling bivariate dependencies represents the second class of EDAs and are implemented by *e.g.,* the mutual information maximization for input clustering (MIMIC) algorithm [16], the combining optimizers with mutual information trees algorithm (COMIT) [17], [18] and the bivariate marginal distribution algorithm (BMDA) [19]. The third class of EDAs can model multivariate variable interactions. Examples of algorithms of this class are the factorized distribution algorithm (FDA) [20], the extended compact genetic algorithm (EcGA) [21] and the Bayesian optimization algorithm (BOA) [22]. It is worth noting that to handle the interaction problem the second and third classes of EDAs require complex learning algorithms and significant additional computational resources. It has been pointed out, for example in [23], that under certain conditions the benefit of this overhead might still be unclear. As a consequence, the first class of EDAs although being simple should not be discredited *a priori*. In this paper the common points and specifics of QEA compared to other EDAs are highlighted. In a similar way, other methods have been also shown to belong to EDAs. For example, as it was shown in [24] and [25], EDAs and the ant colony optimization (ACO) algorithm [26] are actually very similar and differ mainly in the way the probabilistic model is updated.

The use of a probabilistic model is the key concept of any EDAs. The QEA follows the same strategy to guide its search in a given space of solutions. Moreover, in QEA multiple probabilistic models are created and incrementally modified. The idea of using multiple interacting models in EDAs is not new. Probably initiated in [27], this idea is now very popular [28]–[32]. We can identify at least two reasons why the multimodel approach might be useful for optimization problems. First, simple EDAs such as UMDA and PBIL cannot solve complicated problems, which has been shown in [33] and [34]. Second, even advanced EDAs using a complex but still single probabilistic

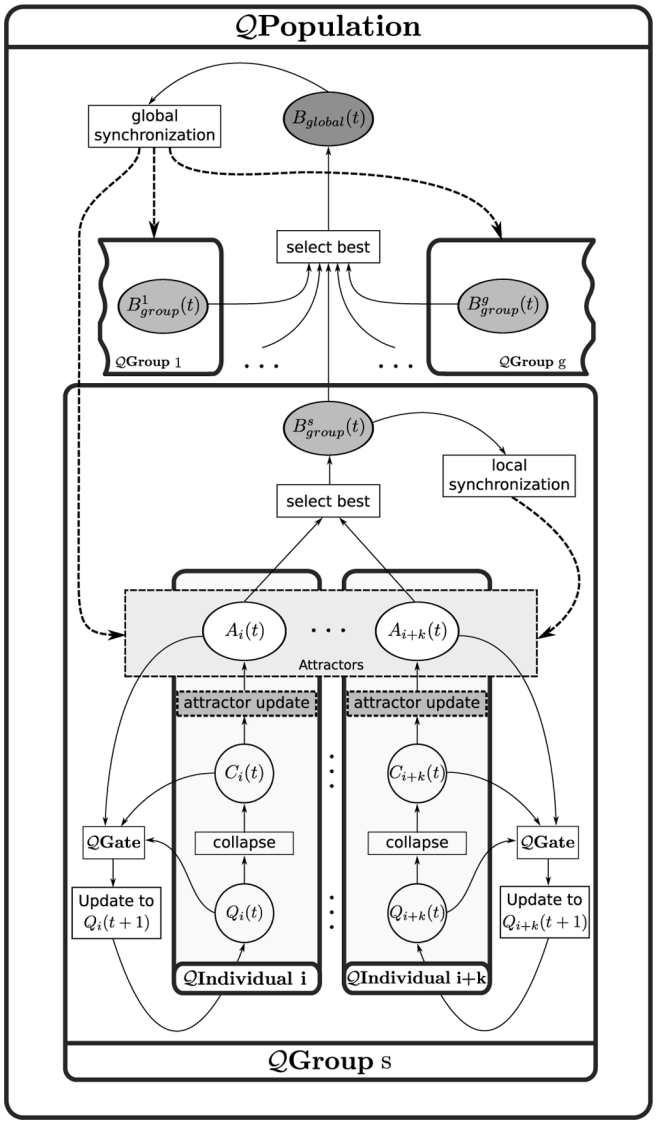

Fig. 1. (v)QEA consists of three different interacting levels: the quantum-individual, -group, and -population levels.

model may not work well in practice [35]. In QEA, the way the probabilistic models interact is unique. It is this interaction that provides the search with an adaptive learning speed and a buffer against potential decision errors. The second aim of this paper is to confirm that several models perform better than only one and then to explain why.

We will begin by briefly reviewing some basic quantum computing principles which we think are necessary to understand the concept of a QEA. After a description of the algorithm itself, we investigate QEA in the light of EDAs. Therefore, the probabilistic model, selection and sampling procedures, learning strategies and population structure used in a QEA are compared to some classical EDAs. In an extensive experimental study, we will investigate the behavior and performance of QEA in terms of fitness, scalability, diversity loss, and robustness against noise. In the final part the role of multiple probabilistic models is discussed and some potential advantages are highlighted.

II. About (V)QEA

The quantum-inspired evolutionary algorithm (QEA) applies quantum computing principles to enhance classical evolutionary algorithms (EAs). We think that the most illustrative example of QEA is the algorithm first proposed by Han and Kim in [6] 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.

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 $\mathcal{Q}$bit [36]. Similar to classical bits, a $\mathcal{Q}$bit may be in "1"-state or "0"-state but additionally also in any superposition of both states. The state $|\Psi\rangle$ of a $\mathcal{Q}$bit $\begin{bmatrix}\alpha\\ \beta\end{bmatrix}$ can be defined as

$$|\Psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{1}$$

where α and β are complex numbers defining probabilities at which the corresponding state is likely to appear when a $\mathcal{Q}$bit is *collapsed*, i.e., read or measured. In another word, the probability of a $\mathcal{Q}$bit to collapse to state "0" and "1" is $|\alpha|^2$ and $|\beta|^2$, respectively.[1] In a more geometrical aspect, a $\mathcal{Q}$bit defines an angle θ 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 [36] for details.

B. Description of the Algorithm

In [37], the authors proposed a revisited description of QEA which we would like to summarize here for the sake of completeness. QEA is a population-based search method. Its behavior can be decomposed in three different and interacting levels; see Fig. 1.

1) Quantum Individuals: The lowest level corresponds to *quantum individuals*[2]. A $\mathcal{Q}$ individual i at generation t contains a $\mathcal{Q}$bit string $Q_i(t)$ and two binary strings $C_i(t)$ and $A_i(t)$. More precisely, Q_i corresponds to a string of N concatenated $\mathcal{Q}$bits

$$Q_i = Q_i^1 Q_i^2 \dots Q_i^N = \begin{bmatrix}\alpha_i^1 & \alpha_i^2 & \dots & \alpha_i^N\\ \beta_i^1 & \beta_i^2 & \dots & \beta_i^N\end{bmatrix}. \tag{2}$$

For the purpose of fitness evaluation, each Q_i is first sampled (or collapsed) to form a binary individual C_i. Each $\mathcal{Q}$bit in Q_i is sampled according to a probability defined by $\left|\beta_i^j\right|^2$, so that C_i represents a configuration in the search space which quality can be classically determined using a fitness function f. In the sense of EA, Q_i is the genotype while C_i is the phenotype of

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

[2]The reader should pay attention that the original notation of Han and Kim has been slightly revisited here. An individual is here composed of $\mathcal{Q}$bit string and two binary strings rather than the $\mathcal{Q}$bit string only.

a given individual. In the sense of EDAs, Q_i defines a probabilistic model

$$\mathcal{P}_i = \left[\left|\beta_i^1\right|^2 \dots \left|\beta_i^N\right|^2\right]$$

while C_i is a realization of this model.

To each individual i, a solution A_i is attached acting as an attractor for Q_i. 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 ($f(A_i) > f(C_i)$ assuming a maximization problem) and if their bit values differ, a quantum gate operator is applied on the corresponding $\mathcal{Q}$bits of Q_i. Thus the probabilistic model P_i defined by Q_i is slightly moved toward the attractor A_i.

The update policy of an attractor A_i can follow two distinctive strategies. In the original QEA [6] an *elitist* update strategy was used, in which the attractor A_i is replaced by C_i only if C_i is better than A_i. In a *nonelitist* update strategy (first introduced in [37]) C_i replaces A_i at every generation. The choice of the update policy has great consequences for the algorithm and changes its behavior completely. To emphasize the importance of the update rule the nonelitist version of QEA has been proposed as *Versatile QEA* (vQEA) [37] as the attractors are able to change every generation and therefore demonstrate a very high volatility. In Section III, we give a more detailed explanation of the role of elitism.

In classical EA, variation operators like crossover or mutation operations are used to explore the search space. The quantum analog for these operators is called a quantum gate. In this study, the rotation gate is used to modify the $\mathcal{Q}$bits. The jth $\mathcal{Q}$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 [7]. We note that the sign of $\Delta\theta$ determines the direction of rotation (clockwise for negative values). In this paper, the application of the rotation gate operator is limited in order to keep θ in the range $[0, \pi/2]$.

2) Quantum Groups: The second level corresponds to *quantum groups*. The population is divided into g $\mathcal{Q}$ groups each containing k $\mathcal{Q}$ 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 $p = g \times k$ $\mathcal{Q}$ individuals forms the *quantum population* and defines the topmost level of QEA. As for the $\mathcal{Q}$ groups, the individuals of the $\mathcal{Q}$ population can synchronize their attractors, too. For that purpose, the best attractor (in terms of fitness) among all $\mathcal{Q}$ 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 $\mathcal{Q}$bits are fixed with $|\alpha|^2 = |\beta|^2 = 1/2$ so that the two states "0" and "1" are equiprobable in collapsed individuals.

C. Previous Results

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 [6], numerical [38], [39] and multi-objective optimization [40], as well as real world problems namely disk allocation method [41], face detection [42], rigid image registration [3], training of multi layer perceptron [43], signal processing [44] and clustering of gene expression data [45].

The quantum-inspired evolutionary algorithm (QEA) introduced in [6] 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 nonoptimal solution is propagated to the global level then this solution starts to attract the entire population. As long as no better solution is found, all the probabilistic models converge towards this global attractor. The probabilistic model becomes unable to produce solutions different from the attractor and therefore QEA can be trapped. Hence, the bad choice of an attractor can quickly become irreversible. QEA is very prone to prematurely converge, suffering mostly by the phenomenon of hitchhiking as experimentally shown in [37].

To prevent the choice of an attractor from being irreversible, the versatile quantum-inspired evolutionary algorithm (vQEA) was proposed in [37], cf. Algorithm 2 in Appendix . In vQEA, 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 periodically shared among the groups or even the whole population. Thus, eventual decision errors do not have long term consequences. vQEA is continuously adapting the search according to local information while the quantum individuals act as memory buffers to keep track of the search history. This leads to a much smoother and more efficient long term exploration of the search space.

In [37], QEA and vQEA have been tested on several benchmark problems and their superiority to a standard genetic algorithm (sGA) was demonstrated. It was claimed that (v) QEA belongs to the class of EDAs. The authors also stated that elitism is not beneficial for that particular type of EDA. This hypothesis was further strengthened by the obtained experimental results. On every benchmark problem vQEA performed better than QEA in terms of speed, solution quality, and scalability. For that particular reason in Sections III–V, we will concentrate our analysis on vQEA only. More generally, we have not been able to found any experimental conditions for which QEA should be preferred to vQEA.

III. VQEA IS AN EDA

According to [15], the algorithms that use a probabilistic model of promising solutions to guide further exploration of the search space are called estimation of distribution algorithms (EDAs). We have seen in Section II-B that each $\mathcal{Q}$ individual define a probability vector and so, as it has already been claimed in [9], [10], and [37], vQEA is a new algorithm belonging to

the class of EDAs. A generic description of EDAs is proposed in Algorithm 1.

Algorithm 1 Estimation of Distribution Algorithm (EDAs)

1: $t \Leftarrow 0$

2: initialize the probabilistic model $\mathcal{P}(t)$

3: **while** not termination condition **do**

4: sample M new solutions from $\mathcal{P}(t)$ into $D(t)$

5: evaluate the elements of $D(t)$

6: select $L \leq M$ solutions from $D(t)$ into $D_s(t)$ using a selection method

7: learn the probabilistic model $\mathcal{P}(t+1)$ from $D_s(t)$ and eventually from $\mathcal{P}(t)$

8: $t \Leftarrow t+1$

9: **end while**

In this section, an extensive study of the features of vQEA is proposed. The common points and specifics of vQEA compared to other EDAs are highlighted.

A. Probabilistic Model

The complexity of the probabilistic model, noted $\mathcal{P}$ in Algorithm 1, varies largely among EDAs. In [11] a survey on EDAs reports three different classes based on the level of interactions between the variables that their models can represent. In the version of vQEA discussed in this paper, binary states are superposed and the eventual interactions between variables are not explicitly taken into account. At the $\mathcal{Q}$ individual level, the probabilistic model $\mathcal{P}_i$, as defined in Section II-B, is a vector of probabilities since each $\left|\beta_i^j\right|^2$ value is used independently for sampling. Therefore, vQEA belongs to the first family of EDAs that assumes independent variables and for which the probabilistic model is a vector of probabilities, such as population-based incremental learning (PBIL) [13], compact GA (cGA) [14] and univariate marginal distribution algorithm (UMDA) [15]; (see Appendix for a detailed description of these algorithms). This family of EDA although being simple should not be discredited *a priori* since the benefit of searching complex variable interactions could, under particular circumstances, be still unclear [23]. We will see in Section V how the p individuals of the $\mathcal{Q}$ population interact to form a multimodel EDA, with $\mathcal{P} = \{\mathcal{P}_1, \ldots, \mathcal{P}_p\}$.

In EDAs, the probabilistic model $\mathcal{P}$ is iteratively updated to account for the fitness of the last L solutions selected in D_s. Nevertheless, the state space on which PBIL, UMDA, cGA, and vQEA act differs. In PBIL, an element $\mathcal{P}^j$ of the probability vector has an arbitrary precision $\Delta\mathcal{P}$ and so the number of possible values for $\mathcal{P}^j$ is infinite. Conversely, in cGA, this number is finite and the precision $\Delta\mathcal{P}$ is constant. The so-called *virtual population size* parameters n determines the accuracy of the model since the update steps have a constant size $\Delta\mathcal{P} = 1/n$. With UMDA, the accuracy of $\mathcal{P}^j$ depends directly on the number L of solutions selected to compute the next probability. However, the update steps are not constant and depends on the variance of the empirical frequency at locus j.

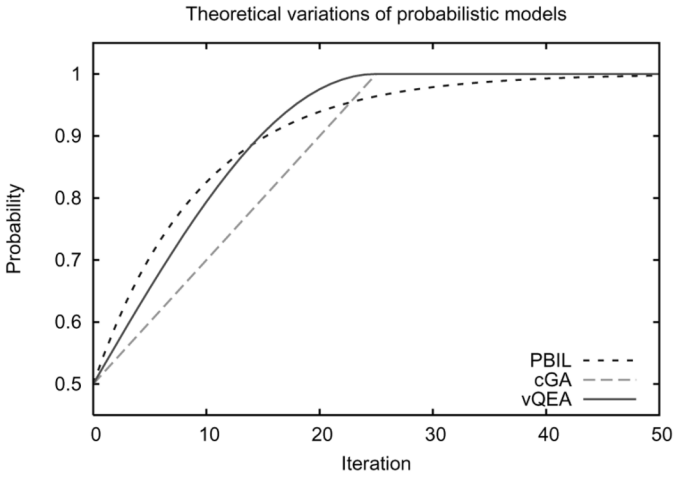

Fig. 2. Theoretical variations of the probabilistic model in PBIL, cGA, and vQEA.

For vQEA, the situation is even more complex. At the level of a $\mathcal{Q}$bit Q_i^j, the application of the rotation gate operator according to $\Delta\theta$ can only produce a finite number $(\pi/2)\times(1/\Delta\theta)$ of positions for the angle $\theta_i^j \in [0, \pi/2]$, and so for the probability $\mathcal{P}_i^j = \left|\beta_i^j\right|^2 = \sin^2\left(\theta_i^j\right)$. The size of the update steps is constant in angle but subsequently varies for $\mathcal{P}_i^j$. More formally we have

$$\begin{aligned}\Delta\mathcal{P}\left(\theta_i^j\right) &= \sin^2\left(\theta_i^j + \Delta\theta\right) - \sin^2\left(\theta_i^j\right) \\ &= 2\cos\left(\theta_i^j\right)\sin\left(\theta_i^j\right) \times \Delta\theta. \end{aligned} \tag{4}$$

It is worth noticing that, according to (4), the more a $\mathcal{Q}$bit is converged (with $\theta_i^j \to (\pi/2)$ or $\theta_i^j \to 0$) the smaller the update step. This phenomenon can be seen as a form of deceleration of the algorithm before convergence.

We can see in Fig. 2 how an element of the probability vector is affected by several successive applications of the update operators for PBIL, cGA, and vQEA. We note that this figure does not reflect the real behavior of the algorithms. This is a theoretical situation where the conditional aspects of the update are not taken into account and so where models are updated at every generation. The initial probability is set to 1/2 and the update direction is toward "1" for each operator. The learning rate of PBIL is fixed to $R_l = 0.1$, the virtual population size of cGA to $n = 50$ and for vQEA the parameter $\Delta\theta$ is equal to $(1/50)(\pi/2)$ and only one $\mathcal{Q}$ individual is used. With such a setting, both cGA and vQEA require 25 update steps to converge.

When considering the population level of vQEA, a set of p probability vectors interact in a complex way (cf.Section V). The accuracy of the overall model $\mathcal{P} = \{\mathcal{P}_1, \ldots, \mathcal{P}_p\}$ can be investigated for example trough looking at the variations of the mean model at locus j, noted $\overline{\mathcal{P}^j}$, such that

$$\overline{\mathcal{P}^j} = \frac{1}{p}\sum_{i=1}^{p}\left|\beta_i^j\right|^2. \tag{5}$$

In vQEA, the update of each θ_i^j and subsequently of each $\left|\beta_i^j\right|^2$ is conditional and is performed independently among the population. Therefore, the number of positions for the average angle $\overline{\theta^j} \in [0, \pi/2]$ is $(\pi/2) \times (1/\Delta\theta) \times (1/p)$.

B. Sampling and Selection

The classical EDAs are distinguished also by the number of solutions M (cf. line 4 of Algorithm 1) sampled at every generation to form the set D. Both PBIL and UMDA require an important number of samples in order to work properly. For example, in [46], the author claimed that M should be large compared to the square root of the problem size N, so that UMDA find the optimum on a One Max problem. Conversely, cGA works with only $M = 2$ bit strings produced per generation. In vQEA, all the $\mathcal{Q}$ individuals collapse during one generation, and so for each Q_i this phase corresponds to the sampling of only one solution from the corresponding model $\mathcal{P}_i$.

After sampling and evaluation of D, the next step in EDAs consists in selecting L solutions into D_s. This subset will be further used during the learning phase. Again, various selection schemes exist in EDAs. For example, PBIL selects only the best (and sometimes together with the worst) element of D[3]. In cGA, a tournament determines a winner and a looser solution, whereas in UMDA a truncation selection is often employed [47], where the $100\alpha\%$ best solutions are selected (typically $\alpha = 1/2$). We note that other models can be used as well such as proportional or tournament selection [48].

At first glance, the selection process of vQEA may appear not so distinctive: as in a tournament each attractor $A_i(t)$ is basically compared in terms of fitness to the last collapsed string $C_i(t)$. Nevertheless, these tournaments are not symmetric. Only if an attractor wins a tournament, a learning phase occurs, otherwise no solution is selected and there will be no learning.

It is noteworthy that $C_i(t+1)$ is sampled from $\mathcal{P}_i(t+1)$ and $A_i(t+1)$ from $\mathcal{P}_i(t)$. If the fitness of $A_i(t)$ is not strictly better that the fitness of $C_i(t)$, then the probabilistic model $\mathcal{P}_i(t)$ stays unchanged, i.e., $\mathcal{P}_i(t+1) = \mathcal{P}_i(t)$. In this case, $C_i(t+1)$ and $A_i(t+1)$ are sampled from the same probabilistic model. Therefore, with an evolutionary point of view, we can consider that they belong to the same generation. Otherwise, after an update of $\mathcal{P}_i(t)$, the selection involves $C_i(t+1)$ and $A_i(t+1)$ issued form generations $t+1$ and t, respectively. In other words, vQEA is a form of *steady-state* EDAs where "parents" and "offspring" may compete against each other. This feature of vQEA is an important specifics since most of the other EDAs are "generational."[4] However, we found some noticeable exceptions where elitism is implemented in EDAs, for example [28] and [50], and so where inter-generational competition exists.

C. Learning and Replacement

The step 7 in Algorithm 1 is a learning phase where the probabilistic model $\mathcal{P}(t+1)$ is built to account for the solutions previously selected in $D_s(t)$. With UMDA, $\mathcal{P}(t+1)$ is fully determined using only the set $D_s(t)$, whereas with PBIL and cGA, both $D_s(t)$ and $\mathcal{P}(t)$ are involved and the learning is incremental. In cGA, the learning is also conditional since the update of the model occurs only at the positions where the winner and the looser bit strings differ. In the original version of PBIL, the learning is unconditional, but we note that some extensions of the basic algorithm have been proposed where the bits of the best and worst solutions are also compared to determine the update; see [13].

Beside the update operator itself (i.e., the rotation gate) the learning process in vQEA is exactly the same as the one employed in cGA. If an attractor A_i wins a tournament, then the binary strings C_i and A_i are systematically compared and the model $\mathcal{P}_i$ is updated toward A_i only where C_i and A_i differ.

We can see in Fig. 3 how an element of the probability vector is affected by the learning process for PBIL, cGA, and vQEA when solving a one bit One Max problem. We note that UMDA is not studied here because it nearly convergences after the first iteration on this problem. Contrary to Fig. 2, we can see the real algorithms working here with the action of the conditional learning for cGA and both the asymmetric selection and conditional learning for vQEA. The curves correspond to the evolution of one probability averaged among 30 independent runs of 200 generations. The learning rate of PBIL is fixed to $R_l = 0.1$ and $M = 2$ solutions are sampled from the model, the virtual population size of cGA is $n = 50$ and for vQEA the parameter $\Delta\theta = (1/50)(\pi/2)$ and only one $\mathcal{Q}$ individual is used. With such a setting, the convergence of PBIL is the fastest mostly because the learning is unconditional. The actual shape is not so different from the theoretical shape reported in Fig. 2. In fact, with only two samples per generation according to this setting of PBIL, the probability of learning a "0" is not null (e.g., 0.25 at the beginning of the run) and so the model is sometimes updated toward the wrong direction slightly slowing down the actual convergence speed. When solving a one bit One Max, the conditional learning prevents the models of cGA and vQEA to be moved toward the wrong direction and also significantly decrease their convergence speed. Besides, the asymmetric selection makes vQEA even slower than cGA. Indeed, the probability of updating the single dimension model $\mathcal{P}$ on this particular problem is $2\mathcal{P}(1-\mathcal{P})$ for cGA and $\mathcal{P}(1-\mathcal{P})$ for vQEA.

Most of the time in vQEA, at generation $t+1$, each $\mathcal{Q}$ individual attractor $A_i(t+1)$ corresponds to the last sampled solution $C_i(t)$. Nevertheless, according to the structure of the $\mathcal{Q}$ population and the local and global synchronization periods, several $\mathcal{Q}$ individuals can also share a common attractor during one generation.

D. Population Structure

Because of the numerous aforementioned specificities of vQEA compared to other EDAs, it is clear that even when considering only a single $\mathcal{Q}$ individual, vQEA is an original EDAs. Nevertheless, what makes vQEA very unique is that it was designed as a coarse-grained algorithm, with a complex structured population of $\mathcal{Q}$ individuals. The situation can be easily compared to multiple demes in EA where subpopulations are artificially separated to promote speciation and where migration allows to share information between demes.

[3]A similar approach has been explored in the best–worst ant system algorithm [24], which also belongs to the class of EDAs.

[4]Some steady-state EDAs exist in the continuous field [49].

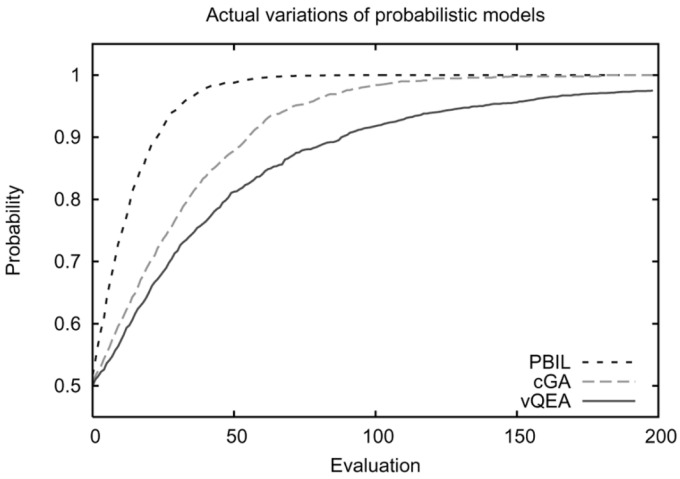

Fig. 3. Actual variations of the probabilistic model in PBIL, cGA, and vQEA.

We note several interesting attempts of multipopulation EDAs [28]–[31].

In vQEA, the structure of the population is fully determined by the number g and the size k of the $\mathcal{Q}$ groups together with the so-called local and global synchronization periods, noted S_{local} and S_{global}, respectively. Actually, there is not a single fixed topology, but rather three superimposed levels of organizations appearing iteratively according to the synchronization periods. As an example, when a global synchronization occurs at time t, the best attractor among the $\mathcal{Q}$ population is selected and then used at time $t+1$ by the $p = g \times k$ $\mathcal{Q}$ individuals. Therefore, at that particular time, the structure of the $\mathcal{Q}$ groups does not matter. The situation is the same for the $\mathcal{Q}$ individuals in a $\mathcal{Q}$ group that are to some extend connected but only during a local synchronization event.

In this paper, we are interested in three different structures: a $\mathcal{Q}$ population containing only one single $\mathcal{Q}$ individual, a $\mathcal{Q}$ population containing a unique $\mathcal{Q}$ group of several $\mathcal{Q}$ individuals and finally the most complex one, a $\mathcal{Q}$ population containing several $\mathcal{Q}$ groups of several $\mathcal{Q}$ individuals.

IV. Experiments

In this section, PBIL, cGA, UMDA, and vQEA are experimentally compared to each other. Besides the fitness performance comparison, we are also interested in the diversity loss, the scalability, and the robustness of each algorithm. However, the performance and the overall behavior of PBIL, cGA, and UMDA strongly depend on the setting of their parameters and the optimal setting varies as a function of the problem to solve. It is not the purpose of this paper to find the most appropriate setting for each algorithm and then to state that one algorithm is better than another.

A. Experimental Setting

We adopted different policies to set the parameters; see Table I. For vQEA, three settings are investigated: a single $\mathcal{Q}$ individual ($\texttt{vQEA}_{1,1}$), one group of ten fully synchronized $\mathcal{Q}$ individuals ($\texttt{vQEA}_{1,10}$) and five groups of 2 $\mathcal{Q}$ individuals synchronized every 100 generations ($\texttt{vQEA}_{5,2}$). For PBIL, we decided to fix M to 10 in such a way that the number of solutions sampled and evaluated in one generation is equivalent to both $\texttt{vQEA}_{1,10}$ and $\texttt{vQEA}_{5,2}$. Actually, according to [46], this setting is suitable for low-dimensional problems ($N \sim 100$). For cGA, the virtual population size is adapted according to the problem size N following the recommendation reported in [51], whereas for UMDA a fixed setting suitable for high-dimensional problems is used.

TABLE I
Parameters Setting

Algorithm	Setting	Name
sGA	$M = 100$, uniform crossover $P_{cross} = 1$, $P_{mut} = 0.01$	sGA
PBIL	$M = 10$, $R_l = 0.1$, $R_m = 0.02$, $R_s = 0.05$	PBIL
cGA	$n = \frac{\sqrt{\pi}}{2}\sqrt{\mathrm{N}}\log \mathrm{N}$	cGA
UMDA	$M = 500$, truncation$= 50\%$	UMDA
vQEA	$g = 1$, $k = 1$, $\Delta\theta = \pi/100$	$\texttt{vQEA}_{1,1}$
-	$g = 1$, $k = 10$, $\Delta\theta = \pi/100$ $S_{global} = 1$	$\texttt{vQEA}_{1,10}$
-	$g = 5$, $k = 2$, $\Delta\theta = \pi/100$ $S_{local} = 1$, $S_{global} = 100$	$\texttt{vQEA}_{5,2}$

The experimental results presented hereafter are obtained by averaging 30 independent runs consisting of 10^5 evaluations for each algorithm and problem tested. We use a statistical unpaired, two-tailed t-test with 95% confidence to determine if results are significantly different.

B. Diversity Loss

The drift phenomenon in EA refers to the loss of genetic diversity due to finite population sampling. In [46], the loss of diversity is studied in the context of EDAs: it is shown that without selection, i.e., on a flat landscape, the variance of the probabilistic model iteratively decays to zero and consequently the model converges towards a fixed configuration. Most EDAs do not compensate for this, and the lost diversity cannot be restored. Moreover, it is also shown that for a nonflat problem the random drift may counteract the effects of selection. Therefore, the parameters of the algorithms have to be tuned properly so that the selection is the main force driving the search.

In this section, an empirical comparison of the loss of diversity of cGA, PBIL, UMDA, and vQEA using the settings reported in Table I is performed on different benchmark problems. Following [46], to estimate the diversity of the bit strings sampled by an EDAs at generation t, we compute the variance v such as

$$v(t) = \sum_{j}^{N} \mathcal{P}^j(t)(1 - \mathcal{P}^j(t)) \tag{6}$$

with $\mathcal{P}^j(t)$ the jth element of the probabilistic model $\mathcal{P}$ at generation t. In the case of vQEA, the average model $\overline{\mathcal{P}}(t)$ over the p $\mathcal{Q}$ individuals (cf. Eq. 5) is used instead. The maximum diversity corresponds to $v_0 = N/4$ and $v(t) = 0$ indicates that the models have converged.

We have seen in Section III-B that the selection process determining the learning phase of vQEA is asymmetric since the update of a model $\mathcal{P}_i$ occurs only if $f(A_i) > f(C_i)$. On a flat landscape, this situation is impossible; therefore, $\mathcal{P}_i$ cannot vary and vQEA cannot loose diversity, i.e., $v(t) = v_0$. We note that this is the optimal behavior for an EDA since if no information is

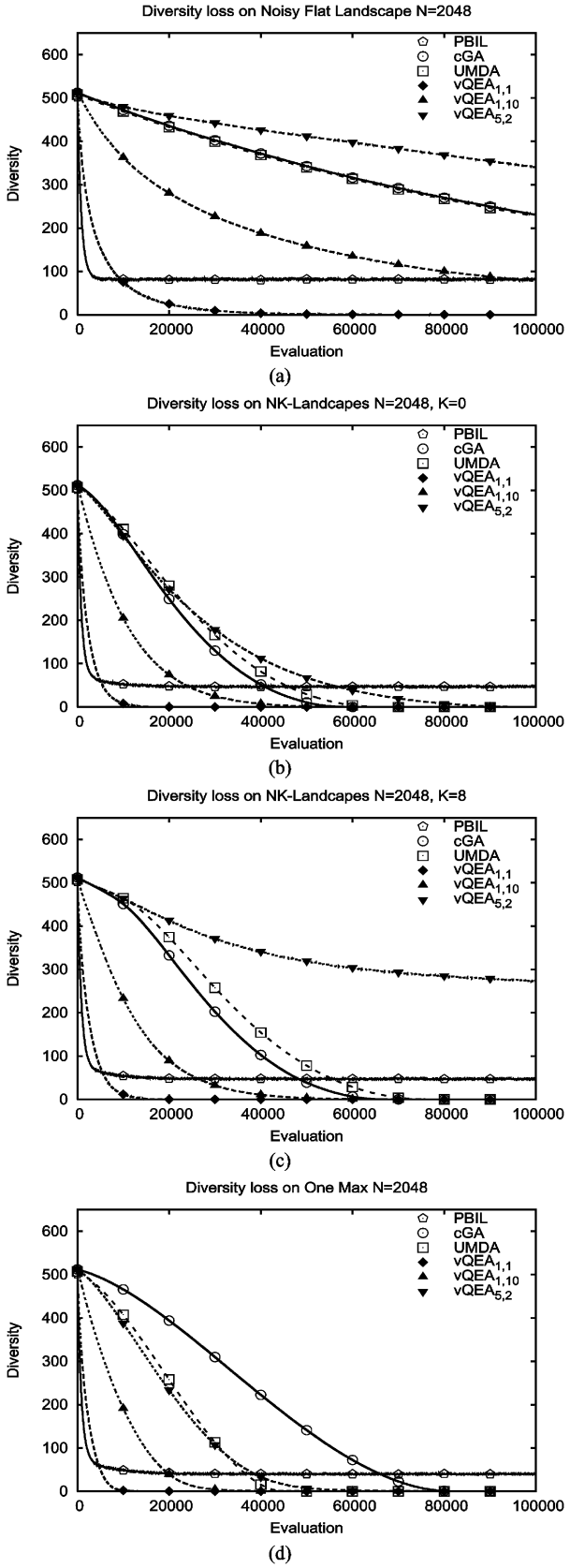

Fig. 4. Loss of diversity on Noisy Flat landscape, One Max, and NK-landscapes with $N = 2048$.

provided each solution of the search space keeps an equal probability of being sampled. The previous remark also stands for the so-called Needle problem. Nevertheless, the drift phenomenon exists in vQEA as well and can be monitored if we add noise to the Flat landscape. We assume a random noise such that the fitness is either 0 or 1 with an equal probability.

In Fig. 4, the average empirical variance $v(t)$ is plotted as a function of the number of fitness evaluations for a Noisy Flat landscape, two NK-landscapes[5] with K equals 0 or 8 and a One Max problem. The size of these four problems is fixed to $N = 2048$ variables.

In the Noisy Flat landscape problem, only random drift can cause the convergence of an algorithm; see Fig. 4(a). PBIL is the algorithm that is the most prone to loose diversity, with $v(t) = v_0/2$ after only 720 evaluations, probably because the setting of the learning rate is not suitable for high-dimensional problems. We see also the effect of the mutation operator of PBIL that perturbs the probabilistic model and guarantees a residual level of diversity around $1/2\,N\,R_m\,R_s(1 - R_s)$, (cf. Appendix for a detailed description of the mutation operator in PBIL). With $L = 250$ for UMDA and $n = 305$ for cGA, the average loss of diversity of both algorithms appears surprisingly almost identical and is also very slow compared to PBIL with $v(t) = v_0/2$ after around 85 000 evaluations. When comparing the loss of diversity of $\texttt{vQEA}_{1,10}$ and $\texttt{vQEA}_{1,1}$, we found that the shapes of the two curves are identical and that only their speed differ. Actually, the loss is exactly ten times faster for $\texttt{vQEA}_{1,1}$ than for $\texttt{vQEA}_{1,10}$, with $v(t) = v_0/2$ after 2400 and 24 000 for $\texttt{vQEA}_{1,1}$ and $\texttt{vQEA}_{1,10}$, respectively. $\texttt{vQEA}_{5,2}$ reports the smallest lost of diversity since after $hbox10^5$ fitness evaluations we still have $v(t) > v_0/2$.

From [53], we know that the mathematical expression of the loss of diversity of UMDA on a Flat landscape is

$$v_{\mathrm{UMDA}}(t) = \frac{N}{4}\left(\frac{1-1}{L}\right)^t. \tag{7}$$

We claim that this expression stands also for the Noisy Flat landscape as defined above. An attempt was made to fit the variance $v(t)$ of vQEA by varying L in (7) for $N = 2048$. It was clear that the loss of diversity of vQEA does not follow the same model as UMDA; nevertheless, the most appropriated values found for L was 65, 160, and 350, for $\texttt{vQEA}_{1,1}$, $\texttt{vQEA}_{1,10}$, and $\texttt{vQEA}_{5,2}$, respectively.

On NK-landscapes, the loss of diversity is due to selection only and the global optimum is unique. For $K = 0$, the N variables can be optimized independently; therefore, this problem is considered to be easy to solve. We see in Fig. 4(b) that the loss of diversity is faster than on the Noisy Flat landscape for each algorithm tested. The convergence of the probabilistic models towards the global optimum is responsible for this loss and except for PBIL, the variance $v(t)$ falls down to zero within the 10^5 evaluations. Apart from $\texttt{vQEA}_{5,2}$, the introduction of interactions between the variables (with $K = 8$) does not seem to affect the way the algorithms convergence. Although, we will see Section V-C that the ten probability vectors of $\texttt{vQEA}_{5,2}$ are all almost converged as well.

[5]According to [52], 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 $K \geq 1$.

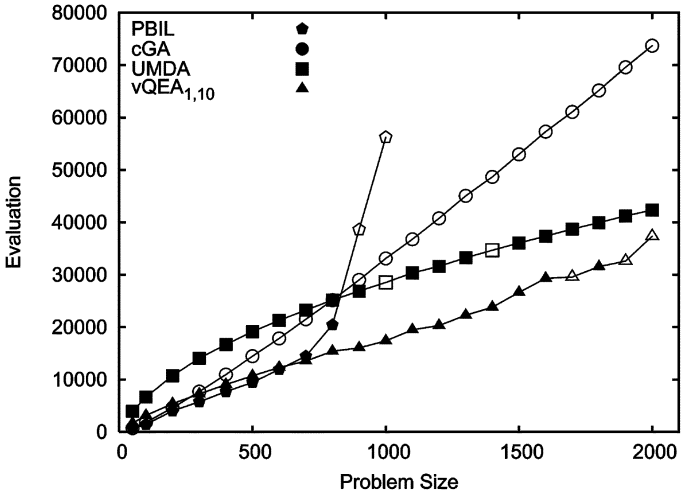

Fig. 5. Number of evaluations as a function of N on the One Max problem.

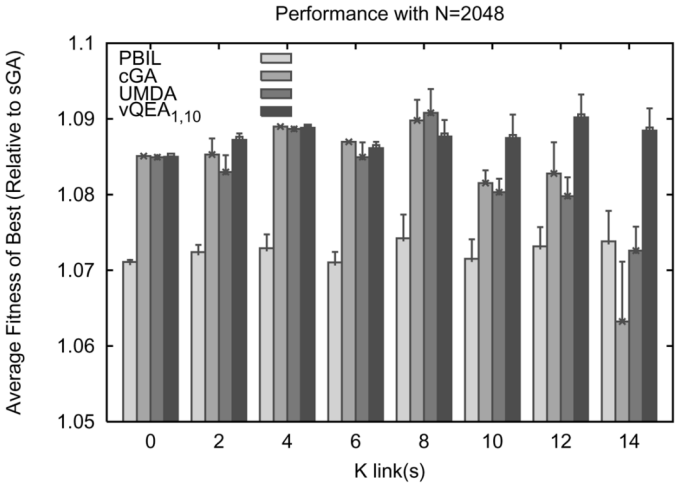

Fig. 6. Performance on NK-landscapes $N = 2048$.

When we rank the algorithms according to the number of evaluation t at which $v(t) = v_0/2$, this rank is identical for the Noisy Flat landscape and NK-landscapes. This is no longer the case on the One Max, in particular for cGA (cf. part (d) of Fig. 4). This problem has no local optima but a single global optimum. Additionally, some neutral dimensions exist as different solutions may have equal fitness values. Hence, both selection and random drift are responsible for the loss of diversity here. As a consequence, the convergence speed of the algorithms is higher on the One Max than on the previously studied problems. Nevertheless, the diversity loss for cGA is slower than on NK-landscapes with $K = 0$. Thus, we can reasonably assume that the neutrality of the problem is responsible for this behavior and of the poor performance of cGA reported in Section V.

C. Scalability

In this section, we investigate the impact of the problem size on the number of fitness evaluations required to find a global optimal solution. For this experiment, we choose the One Max problem as the global optimum is unique and known in advance. Each of the algorithms is applied on the One Max problem with N varying from 50 to 2000 bits. For all the algorithms, the parameter settings reported in Table I were kept unchanged.

Fig. 5 shows the number of fitness evaluations as a function of the problem size N on One Max. For each algorithm, the filled symbol indicates that the global optimum was found in every single run being performed. If only some of the runs were successful, an empty symbol is used instead, and if none, the symbol is not plotted. For small problem sizes, all of the algorithms except of cGA were able to find always the best solution. It is noted that for almost every problem size cGA was unable to find the global optimum in all of the runs. The number of evaluations grows exponentially for PBIL when facing a problem size of $N > 700$. It has to be noted that we have chosen $M = 10$ individuals for PBIL, because of an equal number of evaluations per generation compared to $\text{vQEA}_{1,10}$. For small problem sizes this setting seems to be very suitable: e.g., for $N < 600$ the average fitness evaluations required are the lowest among all other algorithms. We tried other settings for PBIL, but none of them scaled well. For example, using $M = 25$ individuals PBIL performed bad for small problem sizes, but for $N = 1000$ all 30 runs converged to the global optimum, which required on average only 30 138 ($\sigma = 4139.4$) evaluations. In a similar way, the performance of UMDA clearly depends on the problem size. Setting the population size M to 500 is known to be suitable for high-dimensional problems and the results vary as expected: For small size problems, the number of fitness evaluations required is nearly doubled compared to the other algorithms but for $N = 2000$, only $\text{vQEA}_{1,10}$ outperforms UMDA. $\text{vQEA}_{1,10}$ shows an almost linear increase of fitness evaluations while consistently finding the optimal solution up to a problem size of 1600. At least in this study $\text{vQEA}_{1,10}$ demonstrates a high scalability. Furthermore, this single-parameter setting appeared to be suitable for a large variety of problem sizes. The authors were not able to find such a robust setting for any of the other presented algorithms.

D. Fitness

In [37], vQEA was already compared to a standard Genetic Algorithm (sGA) on NK-landscapes and has shown to be superior in terms of both speed and quality of the solution found. In this section, we want to investigate the performance of PBIL, cGA, and UMDA on the same optimization problem. The quality of the results is presented in relation to a sGA: more precisely, the average fitness of the best solutions reached by an algorithm $\mathcal{A}$ is noted $f^*_{\mathcal{A}}$, and the relative performance of $\mathcal{A}$ is defined as the ratio $f^*_{\mathcal{A}}/f^*_{sGA}$. In Fig. 6, the relative performance is plotted for $N = 2048$. It is clearly shown that each EDA outperforms sGA significantly. For small K (and therefore no or low level of interaction between the N variables) PBIL falls behind UMDA, cGA, and $\text{vQEA}_{1,10}$ while the latter three do not show significant difference compared to each other. Nevertheless, it has to be noted that $\text{vQEA}_{1,10}$ shows the lowest variance in the quality of the best solution found among all the other algorithms. With $K \geq 10$, the performance of UMDA and cGA drops significantly obviously being impacted by the higher number of local optima in the fitness landscape. On the other hand, $\text{vQEA}_{1,10}$ stays rather unaffected by the problem

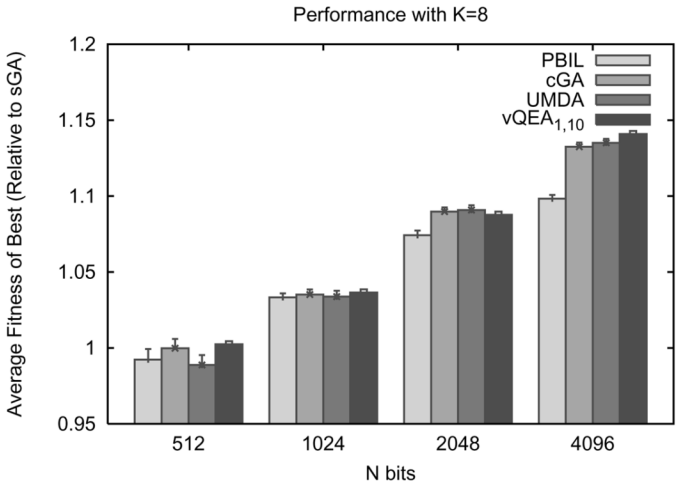

Fig. 7. Performance on NK-landscapes $K = 8$.

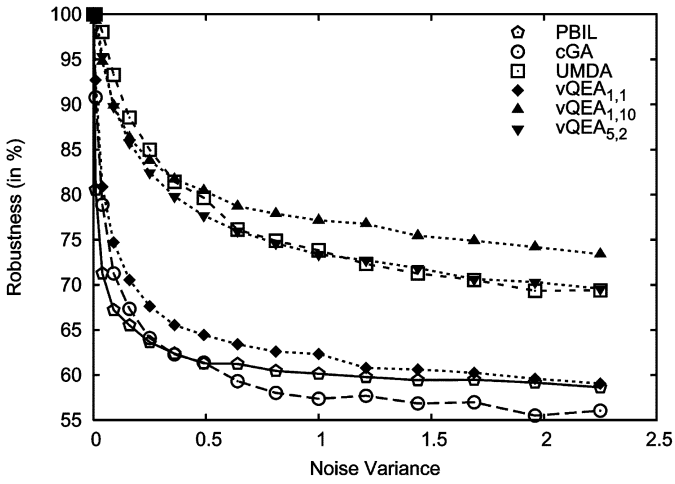

Fig. 8. Robustness as a function of the noise rate on One Max, $N = 256$.

difficulty reporting always between 8% to 9% of fitness higher than a sGA.

Fig. 7 shows the average best relative fitness of several problem sizes N and fixed $K = 8$. For larger problem sizes $N \geq 1024$, each algorithm performs significantly better than a sGA. Again, $\texttt{vQEA}_{1,10}$ shows the lowest variation in the final fitness. There are no significant fitness differences for the problem sizes $N = 512$ and $N = 1024$. For $N \geq 2048$, PBIL falls behind and for $N = 4096$, $\texttt{vQEA}_{1,10}$ delivers the highest solution quality, performing slightly better than each of the other tested algorithms.

E. Robustness

Noise is known to be an important factor that influences evolutionary algorithms. The convergence robustness against fitness noise of PBIL, cGA, UMDA, and vQEA is studied. We assume a multiplicative Gaussian noise, and we define the noisy fitness function F as

$$F(x) = f(x) \cdot \mathcal{N}(1, \sigma^2) \tag{8}$$

with x an element of the solution space and σ^2 the noise variance. Also, we define the robustness $\mathcal{R}(\sigma^2)$ of an algorithm as the ratio between the average best fitness found when noise is applied $(\sigma^2 > 0)$ and the average best fitness found without noise $(\sigma^2 = 0)$. Experiments were performed on One Max with $N = 256$ and $\sigma \in [0, 1.5]$, and the results are plotted in Fig. 8.

For all algorithms we know by construction that $\mathcal{R}(0) = 1.0$, and we see clearly that this robustness is strongly impacted by the increase of the noise variance. Nevertheless, we distinguish two groups of algorithms, with PBIL, cGA, and $\texttt{vQEA}_{1,1}$ on one side and UMDA, $\texttt{vQEA}_{1,10}$, and $\texttt{vQEA}_{5,2}$ on the other side. In the first group, as far as noise is introduced, the robustness decreases extremely fast even for small noise variance. For larger values of noise, the robustness is close to 55% which is comparable to the performance of a random search on a One Max problem. In the second group, the robustness decreases comparatively slower and is still around 70% for $\sigma^2 = 2.25$, where $\texttt{vQEA}_{1,10}$ outperforms all the other algorithms tested with $\mathcal{R}(2.25) = 74\%$.

We note that cGA and $\texttt{vQEA}_{1,1}$ sample, respectively, two and one solutions per generation to update the probability vectors and with PBIL only the best among ten solutions is used. We think that, in the presence of noise, this few number of samples processed leads to decision errors. Indeed, a classical remedy known to counteract the effect of noise in EA is to perform multiple evaluations of the fitness. With UMDA $M = 500$ solutions are analyzed before a learning phase occurs. This large number of evaluations before a model update is probably responsible for the convergence towards an average good solution. Population-based search algorithms are also known to be robust because of their self-averaging nature. We claim that in vQEA the $\mathcal{Q}$ population acts as a buffer against decision errors because $\mathcal{Q}$ individuals are able to share information about the search space. Since in $\texttt{vQEA}_{1,10}$, all $\mathcal{Q}$ individuals are embedded in the same $\mathcal{Q}$ group and thus follow the same attractor $(S_{\text{local}} = 1)$, the information share is maximized and therefore $\texttt{vQEA}_{1,10}$ is the most robust of the algorithms tested here. Moreover, we know from [54] that the interactions between variables may be seen as a form of fitness noise by the algorithms what could also explain the good results reported in the previous section for $\texttt{vQEA}_{1,10}$ on NK-landscapes for high degrees of epistasy $(K \geq 10)$.

V. Role of Multiple Models

In this section, we investigate further the role of the set of multiple probabilistic models $\mathcal{P} = \{\mathcal{P}_1, \ldots, \mathcal{P}_p\}$ in vQEA.

A. Do Multiple Models Perform Better Than Only One?

vQEA has been originally introduced as a coarse-grained evolutionary algorithm with several interacting $\mathcal{Q}$ individuals. Nevertheless, we are not aware of any serious demonstration of the superiority of using a $\mathcal{Q}$ population compare to using only a single $\mathcal{Q}$ individual. A fair comparison between $\texttt{vQEA}_{1,1}$ and $\texttt{vQEA}_{1,10}$, i.e., not based on an equivalent number of generations but on a equivalent number of fitness evaluations, is performed on One Max and NK-landscapes problems. For all the experiments carried out, the fitness of the best solution found with $\texttt{vQEA}_{1,10}$ is better or equal to the best solution produced with $\texttt{vQEA}_{1,1}$. As an illustration, in Fig. 9, the average evolution of the best fitness found on NK-landscapes with $N = 256$ and $K = 16$

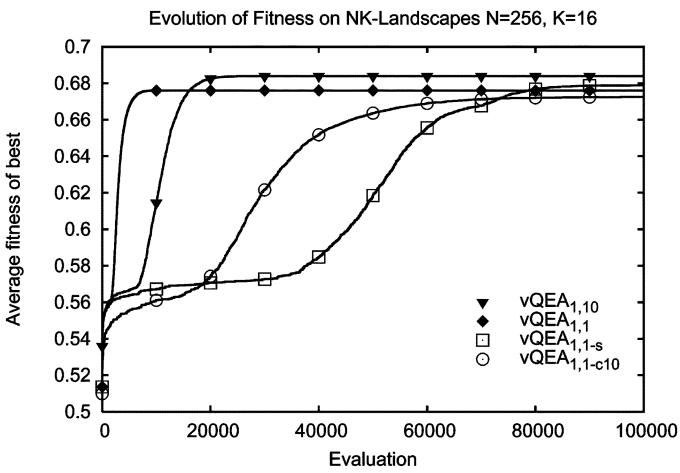

Fig. 9. Fitness evolution of single and multiple models vQEA on NK-landscapes.

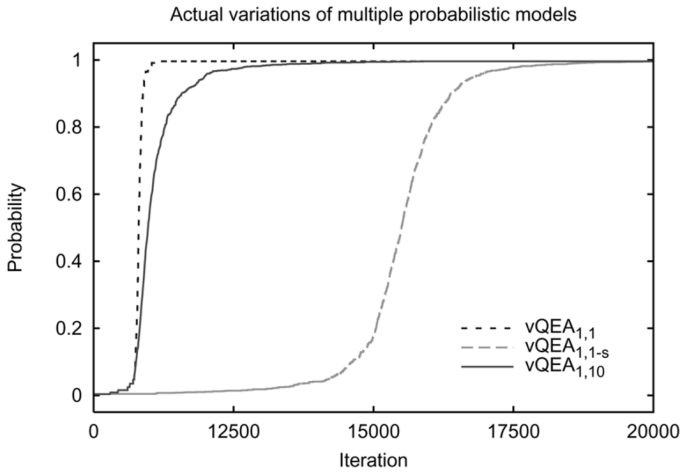

Fig. 10. Actual variations of the mean probabilistic model in vQEA.

is plotted for $\texttt{vQEA}_{1,1}$ and $\texttt{vQEA}_{1,10}$ as a function of the number of evaluations. The setting of the parameters is given in Table I. For both settings, the fitness improves extremely fast after few evaluations then, while $\texttt{vQEA}_{1,1}$ keeps a similar trend until it prematurely convergences, $\texttt{vQEA}_{1,10}$ reports a more step-wise increase and finally reaches a higher fitness level.

In $\texttt{vQEA}_{1,10}$, ten $\mathcal{Q}$ individuals synchronize their attractors at every generation t using the best solution sampled at generation $t-1$. This setting implies that the ten corresponding probability vectors $\mathcal{P}_1,\ldots,\mathcal{P}_{10}$ are all following a unique attractor and therefore the same direction in the search space. If we assume that each model $\mathcal{P}_i(t)$ is not so different from the mean model $\overline{\mathcal{P}}(t)$, having several models instead of only one may appear *a priori* useless. Nevertheless, the benefit of using multiple models is clearly demonstrated experimentally. Hence, we investigate two different hypothesis to explain the better results obtained with $\texttt{vQEA}_{1,10}$.

In the first hypothesis, we assume that $\texttt{vQEA}_{1,10}$ benefits from the fact that the search direction is chosen after sampling ten solutions, i.e., one per model. For that reason, we propose to produce ten solutions from the single probabilistic model and then to use the best among them as the next attractor. This algorithm is noted $\texttt{vQEA}_{1,1\text{-}c10}$ in Fig. 9. We see that $\texttt{vQEA}_{1,1\text{-}c10}$ is outperformed by $\texttt{vQEA}_{1,1}$ and $\texttt{vQEA}_{1,10}$ in terms of speed and average fitness of the best solution found.

In the second hypothesis, we assume that $\texttt{vQEA}_{1,10}$ benefits from a slower convergence speed. We note that in $\texttt{vQEA}_{1,10}$, it may happen that only one vector $\mathcal{P}_i$ among the ten is updated during one generation t. In that case, the average model $\overline{\mathcal{P}}(t)$ moves very slowly towards the attractor, and the update steps correspond to $\Delta\theta/10$. Therefore, we propose to evaluate the performance of a single $\mathcal{Q}$ individual vQEA with a ten times smaller update step $\Delta\theta = 1/10 \times \pi/100$. As expected with this setting the algorithm noted $\texttt{vQEA}_{1,1\text{-}s}$ in Fig. 9 reports a slower convergence speed and outperforms $\texttt{vQEA}_{1,1}$ in terms of fitness. The fitness increases slowly, in a step-wise manner similar to $\texttt{vQEA}_{1,10}$ but finally reaches a significantly smaller fitness value.

We have gone to great effort to reproduce results similar to $\texttt{vQEA}_{1,10}$ using one probabilistic model only but we have not been successful. Therefore, we claim that even when they are fully synchronized and so almost equal, the multiple probabilistic models perform better.

B. Adaptive Learning Speed

We believe that the interplay of the fully synchronized multiple models leads to an adaptive learning speed. To illustrate this we plot the evolution of the mean model $\overline{\mathcal{P}}(t)$ when solving a one bit One Max problem for $\texttt{vQEA}_{1,1}$, $\texttt{vQEA}_{1,1\text{-}s}$, and $\texttt{vQEA}_{1,10}$; see Fig. 10. For that specific experiment, the initial probability is set to $\sin^2(\Delta\theta) \sim 0$. The only difference between $\texttt{vQEA}_{1,1}$ and $\texttt{vQEA}_{1,1\text{-}s}$ is the setting of $\Delta\theta$ and as a consequence their convergence speeds. We see that for these two settings, the evolution of the probability looks like a $\arctan$ function. In particular, the shape of the two curves is identical when the probability leaves zero or when it reaches one. On the contrary for $\texttt{vQEA}_{1,10}$ an asymmetric behavior is observed: the average probability leaves zero much more faster than it reaches one. More precisely, the average probability evolves in a way similar to $\texttt{vQEA}_{1,1}$ when moving away from zero and then similar to $\texttt{vQEA}_{1,1\text{-}s}$ when approaching one. This is a very desired behavior as we expect that the algorithm dedicates more efforts to explore the promising areas of the search space.

This phenomenon can easily be explained when considering the set of ten vectors $\{\mathcal{P}_1,\ldots,\mathcal{P}_{10}\}$. At the beginning of this experiment, almost every solution C_i produced is "0." When by chance a "1" is sampled, it becomes the next attractor for the $\mathcal{Q}$ population and so there is a high probability that the ten models are updated at the same time. Therefore, the learning speed of $\overline{\mathcal{P}}(t)$ can be high, i.e., depending on $\Delta\theta$. Afterwards, the number of models updated during one generation starts to decrease. The extreme case is when only one model is updated what implies a much more slower learning speed for $\overline{\mathcal{P}}(t)$, i.e., corresponding to $1/10 \times \Delta\theta$. The situation can be seen as a voting mechanism controlling the overall learning speed. When the $\mathcal{Q}$ individuals all agree that a direction in the search space is not appropriate, their models all move away and subsequently the average model moves fast. Conversely when they disagree, the mean model moves very slowly giving more accuracy and therefore more time to the algorithm to take the right decision.

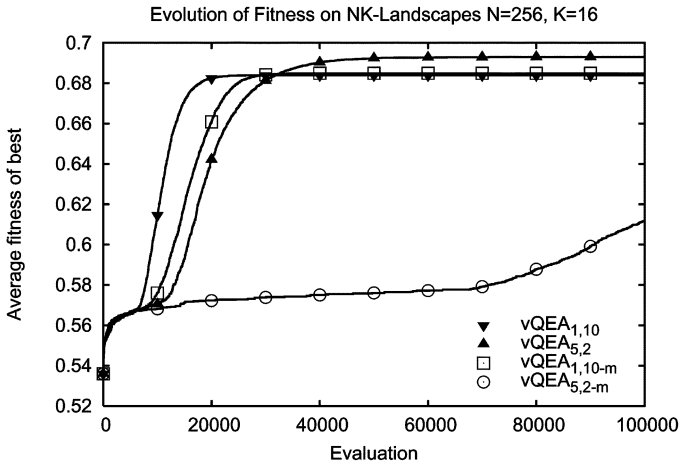

Fig. 11. Fitness evolution of mean and multiple models vQEA on NK-landscapes.

We believe that this adaptive learning speed is also responsible for the good results reported for vQEA in terms of robustness to fitness noise.

C. Do Multiple Models Perform Better Than a Mean Model?

In the two previous sections, it is assumed that in $\texttt{vQEA}_{1,10}$ the models $\mathcal{P}_1, \ldots, \mathcal{P}_{10}$ are almost identical at time t and therefore equivalent to the mean model $\overline{\mathcal{P}}(t)$. Subsequently, it is assumed that the distribution of solutions in the set $\{C_1, \ldots, C_{10}\}$ sampled from the ten models at time t is to some extend equivalent to the distribution obtained when sampling ten solutions from $\overline{\mathcal{P}}(t)$. We now evaluate the validity of this assumption for $\texttt{vQEA}_{1,10}$ but also for $\texttt{vQEA}_{5,2}$.

For that purpose, we introduce two variants, noted $\texttt{vQEA}_{1,10\text{-}m}$ and $\texttt{vQEA}_{5,2\text{-}m}$, where the mean model is used for sampling. More precisely, the overall structure and settings of the algorithm are kept unchanged except that a mean model $\overline{\mathcal{P}}(t)$ is computed every generation and then used to produce the individual solutions $C_i(t)$. In particular, it is noteworthy that the adaptive learning speed described earlier works for these two variants as well and therefore any noticeable variation in the performance of the algorithm may result from the use of the mean model only. In Fig. 11, the average evolution of the best fitness found on NK-landscapes with $N = 256$ and $K = 16$ is plotted as a function of the number of evaluations.

We note that the two curves obtained for $\texttt{vQEA}_{1,10}$ and $\texttt{vQEA}_{1,10\text{-}m}$ are very similar and their average final fitness values are statistically identical. Notwithstanding, the slightly faster convergence of $\texttt{vQEA}_{1,10}$ compare to $\texttt{vQEA}_{1,10\text{-}m}$, the assumption made in the previous section seems to hold on this problem: sampling ten fully synchronized models $\mathcal{P}_i$ is indeed comparable to sample the corresponding mean model $\overline{\mathcal{P}}$. The situation is clearly not the same for $\texttt{vQEA}_{5,2}$. While $\texttt{vQEA}_{5,2}$ is the best setting of vQEA tested on this problem, $\texttt{vQEA}_{5,2\text{-}m}$ reports extremely poor results. Therefore, we claim that when they are not fully synchronized the multiple probabilistic models can perform better than the mean model.

In $\texttt{vQEA}_{5,2}$, five $\mathcal{Q}$ groups of two fully synchronized $\mathcal{Q}$ individuals are evolved and the best attractor among the groups is shared, according to the parameter S_{global}, every 100th generation. However, in vQEA the attractor are systematically replaced at every generation, so that a given synchronization can affect the evolution of the $\mathcal{Q}$ groups during a single generation only. As a consequence the groups can evolve almost separately towards different regions of the search space. With $K = 16$ epistatic links in the problem, the interactions between the 256 variables are important and the problem is not easy to solve. We believe that with $\texttt{vQEA}_{5,2}$ each $\mathcal{Q}$ group can specialize on different patterns of bits and so that the multiple models of vQEA allow to sample a more complex distribution of solutions than with a single probability vector.

D. Measuring Diversity

In order to measure the diversity of the solutions sampled by the multiple models in vQEA, the variance $v(t)$, as defined in (6), is not necessarily adapted. Actually, in Section IV-B, the variance was computed using the mean model $\overline{\mathcal{P}}(t)$, but clearly this procedure does not consider the conditional probabilities and is not sufficient to represent interactions among variables. Thus, the more the vectors $\mathcal{P}_1, \ldots, \mathcal{P}_p$ differ at time t, the less the variance $v(t)$ is suitable. Hence, we propose another approach where two metrics are used to represent the diversity of the solutions produced at time t: the convergence of the $\mathcal{Q}$ population noted $\text{Conv}(t)$ and the pairwise distance between the $\mathcal{Q}$ individuals noted $\text{Dist}(t)$.

The convergence of a $\mathcal{Q}$ population reflects how the N $\mathcal{Q}$bits have converged in the whole population. We define Conv^j, the $\mathcal{Q}$bit convergence at locus j

$$\text{Conv}^j = \frac{2}{p} \sum_{i}^{p} \left| \mathcal{P}_i^j - \frac{1}{2} \right| \tag{9}$$

and so the convergence of the $\mathcal{Q}$ population corresponds to the mean $\mathcal{Q}$bit convergence over N $\mathcal{Q}$bits such that

$$\text{Conv} = \frac{1}{N} \sum_{j}^{N} \text{Conv}^j. \tag{10}$$

The pairwise distance between the $\mathcal{Q}$ individuals reflects how their probabilistic models differ. To represent the distance $\text{Dist}_{i,k}$ between two probability vectors $\mathcal{P}_i$ and $\mathcal{P}_k$, we propose to simply compute

$$\text{Dist}_{i,k} = \frac{1}{N} \sum_{j}^{N} \left| \mathcal{P}_i^j - \mathcal{P}_k^j \right|. \tag{11}$$

Following [55], this metric can be easily interpreted as the proportion of mutational changes required to transform a set of solutions sampled from $\mathcal{P}_i$ to a set of solutions sampled form $\mathcal{P}_k$. Hence, the pairwise distance between p $\mathcal{Q}$ individuals corresponds to

$$\text{Dist} = \frac{2}{p(p-1)} \sum_{i=1}^{p} \sum_{k=i+1}^{p} \text{Dist}_{i,k}. \tag{12}$$

We have computed the evolution of $\text{Conv}(t)$ together with $\text{Dist}(t)$ on NK-landscapes with $N = 2048$ for $K = 0$ and $K = 8$. The setting was $\texttt{vQEA}_{5,2}$ as described in Table I, i.e.,

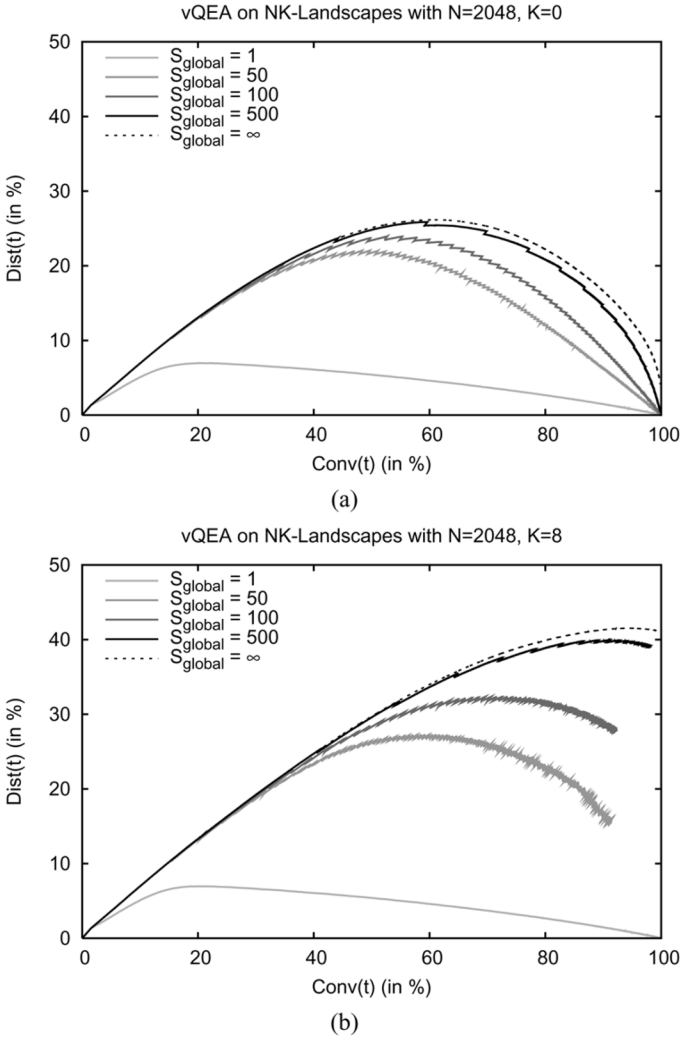

Fig. 12. Pairwise Distance between $\mathcal{Q}$ individuals versus Convergence of the $\mathcal{Q}$ population on NK-landscapes.

five $\mathcal{Q}$ groups of two synchronized $\mathcal{Q}$ individuals are evolved and the best attractor is shared according to S_{global}. Furthermore, the influence of the global synchronization period S_{global} was also investigated. The results averaged over 30 independent runs of 10^5 evaluations are plotted in Fig. 12.

For every curve, a common trend is reported. After the initialization phase, each $\mathcal{Q}$ individual defines a probability vector $\mathcal{P}_i$ which elements are all set to 1/2 and therefore Conv(0) and Dist(0) are both equal to 0. At that particular time, the diversity of the solutions sampled is maximum. Then, under the effects of selection (together with drift on neutral problems), the $\mathcal{Q}$ population starts to converge with Conv$(t) > 0$ and the probabilistic models becomes more and more different until Dist(t) reaches a maximum. Finally, the pairwise distance decreases while the $\mathcal{Q}$ population keeps converging continuously. As expected, S_{global} determines the amplitude of the peak of maximum distance between the multiple models. With $S_{\text{global}} = 1$, the models are fully synchronized (as in $\texttt{vQEA}_{1,10}$). For both $K = 0$ and $K = 8$, the maximum value for Dist(t) with $S_{\text{global}} = 1$, is approximately 7%. This very low value means that the multiple models represent subspaces (hypercubes) that differ by 7% of their bits. With higher values for S_{global}, the $\mathcal{Q}$ groups are more likely to evolve towards different regions of the search space and the maximum value for Dist(t) increases.

When the multiple models are not fully synchronized, i.e., $S_{\text{global}} > 1$, we note major discrepancies between the case $K = 0$ and $K = 8$, cf.Fig. 12(a) and (b), respectively.

For $K = 0$, the maximum value for Dist(t) is around 25% when the attractors are never synchronized ($S_{\text{global}} = \infty$), i.e., when the five $\mathcal{Q}$ groups evolve separately. After 10^5 evaluations we have Conv$(t) = 1$ and Dist$(t) = 0$, even for $S_{\text{global}} = \infty$. In this situation, the five $\mathcal{Q}$ groups have converged towards the same solution of the search space. We observe also a sawtooth shape of the curves where each tooth corresponds to an episode of synchronization of the attractors. Inasmuch as with $K = 0$ there is no local optima, the information carried by the attractors is not contradictory and therefore is smoothly exchanged between the $\mathcal{Q}$ groups.

For $K = 8$, the maximum value for Dist(t) is around 40% for $S_{\text{global}} = \infty$. After 10^5 evaluations, Dist(t) is not equal to zero and for $S_{\text{global}} = 50$, 100, and 500, the $\mathcal{Q}$ population has not converged. The sawtooth shape is no more reported here and instead the curves are very rugged, in particular $S_{\text{global}} = 50$. We believe that with $K = 8$, the information carried by attractors can be contradictory and therefore not easily exchanged between the $\mathcal{Q}$ groups what tends to slow down the convergence speed of the $\mathcal{Q}$ population. Nevertheless, as long as the best performance in terms of fitness is obtained for $S_{\text{global}} = 100$, some information is exchanged through the synchronization process. This is the reason why one of the role of the multiple models in vQEA is to allow a more diverse exploration of the search space than with a single model only.

VI. Conclusion

Behind the quantum metaphor, vQEA is an original approach that belongs to the class of EDAs. It clearly shares some common features with several simple EDAs such as PBIL and cGA, but its performance is more similar to the one of UMDA, for example what concerns the loss of diversity, the scalability and the robustness to fitness noise. Therefore, we believe that vQEA should greatly benefit from former work about simple EDA where interactions between variables are not taken into account.

The main specific feature of the vQEA is obviously the multimodel approach. In this paper, the advantages of manipulating several probability vectors instead of only one are empirically demonstrated. First, vQEA is an effective algorithm that works with fairly generic settings of the control parameters for a collection of benchmark problems of various sizes, with different levels of interactions between variables and numbers of neutral dimensions. It is worth noticing that in this study, no particular efforts have been dedicated for finding the best possible settings of vQEA but rather we used a setting directly borrowed from previous work on QEA, which behavior is quite dissimilar to vQEA, and on a different test problem. Second, the $\mathcal{Q}$ population allows to buffer against a finite number of decision errors what makes vQEA robust against fitness noise. Finally, we have shown that vQEA can perform better than other simple

EDAs when links are introduced between variables. Those interesting results about the multimodel approach in vQEA can be explained by two main reasons: on the one hand, an adaptive learning speed and on the other hand a more diverse sampling of the search space than other EDAs with a single probability vector. Future work should compare the mechanisms of existing multimodel EDAs approaches [27]–[32] to the one used in vQEA and evaluate their relative performance empirically.

The way the $\mathcal{Q}$ population is structured, i.e., number and size of the $\mathcal{Q}$ groups together with the local and global synchronization periods, directly controls the adaptive learning speed and the diversity of the solutions sampled by vQEA. To properly choose this structure, we suggest the following approach. First of all, the $\mathcal{Q}$ individuals should be fully synchronized in a $\mathcal{Q}$ group (with $S_{\text{local}} = 1$) of size k in such a way that k determines the variation of the learning speed from $\Delta\theta/k$ to $\Delta\theta$. In a second time, several $\mathcal{Q}$ groups should be introduced as long as the problem is known to report a significant number of local optima or similarly a significant level of dependency between the variables. Then the global synchronization period controlling the diversity of the sampled solutions can be set inversely to the size of the problem.

Despite the scalability of vQEA, the generic setting proposed in this study is probably not optimal, and therefore a general expression should be proposed. In particular, the optimal setting of $\Delta\theta$ according the size of the problem is still unknown and inasmuch as $\Delta\theta$ gives the fastest learning speed, its setting should be investigated, at least empirically, for example on a simple One Max problem.

We have seen that one of the strength of vQEA comes from the specialization of $\mathcal{Q}$ groups on diverse subspaces. Actually, only the stochastic behavior of the $\mathcal{Q}$ individuals driven by fitness selection makes the $\mathcal{Q}$ groups to converge towards different regions of the search space. So far, even with a very low synchronization frequency, we cannot guarantee the diversity of the sampling for every problems. This question should be discussed so that extra mechanisms for ensuring specialization can eventually be added.

The impact of the synchronization events on the probabilistic models has been shown to be rather limited. Nevertheless, the synchronization of attractors definitively helps the multiple $\mathcal{Q}$ individuals to optimize nondecomposable problems. So far, the extend to which these problems can be solved using vQEA remains unclear. From the experiments presented in this paper, we believe that the performance of vQEA lies somewhere in between the one reported by the simple and the complex EDAs. Therefore, the efficiency of vQEA in terms of exchange of information and building blocks mixing should be addressed in future work, for example using a flexible benchmark such as the Random Additively Decomposable Problems [56], where the variable interactions can be explicitly controlled and additionally the global optimum is known.

APPENDIX
DESCRIPTION OF ALGORITHMS

Algorithm 2 vQEA

1: initialize each Q_i
2: initialize each A_i
3: **while** not termination condition **do**
4: **for all** $i \in [1, p]$ **do**
5: sample 1 new solution C_i from Q_i
6: evaluate C_i
7: **if** $f(A_i)$ **better than** $f(C_i)$ **then**
8: learn_model (A_i, C_i, Q_i)
9: **end if**
10: $A_i \leftarrow C_i$
11: **end for**
12:check local and global synchronization
13: **end while**
14:
15: **function** learn_model (A_i, C_i, Q_i)
16: **for all** $j \in [1, N]$ **do**
17: **if** $A_i^j \neq C_i^j$ **then**
18: **if** $A_i^j = 1$ **then**
19: $Q_i^j \leftarrow$ rotate Q_i^j towards A_i^j using $\Delta\theta$
20: **else**
21: $Q_i^j \leftarrow$ rotate Q_i^j towards A_i^j using $-\Delta\theta$
22: **end if**
23: **end if**
24: **end for**

Algorithm 3 PBIL

1: initialize the probabilistic model $\mathcal{P}$
2: **while** not termination condition **do**
3: sample M new solutions from $\mathcal{P}$ into D
4: evaluate the elements of D
5: select $best$ from D
6: **for all** $j \in [1, N]$ **do**
7: $\mathcal{P}^j \leftarrow \mathcal{P}^j \times (1.0 - R_l) + best^j \times R_l$
8: **if** $rand(0, 1] < R_m$ **then**
9: $\mathcal{P}^j \leftarrow \mathcal{P}^j \times (1.0 - R_s) + rand(0.0 \textbf{ or } 1.0) \times R_s$
10: **end if**
11: **end for**
12: **end while**

Algorithm 4 cGA

1: initialize the probabilistic model $\mathcal{P}$
2: **while** not termination condition **do**

3: sample 2 new solutions from $\mathcal{P}$ into D

4: evaluate the elements of D

5: select $winner$ and $loser$ from D

6: learn_model $(winner, loser, \mathcal{P})$

7: **end while**

8:

9: **function** learn_model $(winner, loser, \mathcal{P})$

10: **for all** $j \in [1, N]$ **do**

11: **if** $winner^j \neq loser^j$ **then**

12: **if** $winner^j = 1$ **then**

13: $\mathcal{P}^j \leftarrow \mathcal{P}^j + 1/n$

14: **else**

15: $\mathcal{P}^j \leftarrow \mathcal{P}^j - 1/n$

16: **end if**

17: **end if**

18: **end for**

Algorithm 5 UMDA

1: initialize the probabilistic model $\mathcal{P}$

2: **while** not termination condition **do**

3: sample M new solutions from $\mathcal{P}$ into D

4: evaluate the elements of D

5: select $L = \alpha * M$ solutions from D into D_s

6: **for all** $j \in [1, N]$ **do**

7: $\mathcal{P}^j \leftarrow$ compute marginal frequency at locus i in D_s

8: **end for**

9: **end while**

REFERENCES

[1] A. Narayanan and T. Menneer, "Quantum artificial neural network architectures and components," *Inf. Sci.*, pp. 231–255, 1995.

[2] A. Narayanan and M. Moore, "Quantum inspired genetic algorithms," in *Proc. Int. Conf. Evol. Comput.*, 1996, pp. 61–66.

[3] A. Draa, M. Batouche, and H. Talbi, "A quantum-inspired differential evolution algorithm for rigid image registration," *Trans. Eng., Comput., Technol.*, pp. 408–411, 2004.

[4] Y. Li and L. Jiao, "Quantum-inspired immune clonal algorithm," in *Proc. ICARIS*, 2005, pp. 304–317.

[5] J. Liu, J. Sun, and W. Xu, "Improving quantum-behaved particle swarm optimization by simulated annealing," in *Proc. ICIC (3)*, 2006, pp. 130–136.

[6] K.-H. Han and J.-H. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *IEEE Trans. Evol. Comput.*, vol. 6, pp. 580–593, Dec. 2002.

[7] K.-H. Han and J.-H. Kim, "On setting the parameters of quantum-inspired evolutionary algorithm for practical application," *Proc. IEEE Congr. Evol. Comput.*, vol. 1, pp. 178–194, 2003.

[8] K.-H. Han, "Quantum-inspired evolutionary algorithm," Ph.D. dissertation, Korea Adv. Inst. Sci. Technol. (KAIST), Daejeon, Korea, 2003.

[9] S. Zhou and Z. Sun, "A new approach belonging to EDAs: Quantum-inspired genetic algorithm with only one chromosome," in *Proc. ICNC (3)*, 2005, pp. 141–150.

[10] K.-H. Han and J.-H. Kim, "On the analysis of the quantum-inspired evolutionary algorithm with a single individual," *Proc. IEEE Congr. Evolut. Comput.*, pp. 16–21, 2006.

[11] M. Pelikan, D. Goldberg, and F. Lobo, "A survey of optimization by building and using probabilistic model," 1999, IlliGAL, Tech. Rep. 99018.

[12] P. Larrañaga, E. Etxeberria, J. A. Lozano, and J. M. Peña, Optimization by learning and simulation of Bayesian and Gaussian networks Univ. of the Basque Country, 1999, Tech. Rep. EHU-KZAA-4/99.

[13] S. Baluja, Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning Carnegie Mellon Univ., Pittsburgh, PA, 1994, Tech. Rep. CMU-CS-94-163.

[14] G. R. Harik, F. G. Lobo, and D. E. Goldberg, "The compact genetic algorithm," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 287–297, Nov. 1999.

[15] H. Mühlenbein and G. Paass, "From recombination of genes to the estimation of distributions i. binary parameters," in *Proc. Int. Conf. Evol. Comput., Parallel Problem Solving From Nature—PPSN IV*, 1996, pp. 178–187.

[16] J. S. de Bonet, C. L. Isbell, Jr, and P. Viola, "MIMIC: Finding optima by estimating probability densities," in *Advances in Neural Information Processing Systems*, M. C. Mozer, M. I. Jordan, and T. Petsche, Eds. Cambridge, MA: MIT Press, 1997, vol. 9, pp. 424–424.

[17] S. Baluja and S. Davies, "Using optimal dependency-trees for combinatorial optimization: Learning the structure of the search space," in *Proc. 1997 Int. Conf. Mach. Learn.*, 1997, pp. 30–38.

[18] S. Baluja and S. Davies, "Fast probabilistic modeling for combinatorial optimization," in *Proc. 15th National Conf. Artif. Intell. (AAAI-98)*, 1998, pp. 469–476.

[19] M. Pelikan and H. Mühlenbein, "The bivariate marginal distribution algorithm," in *Advances in Soft Computing—Engineering Design and Manufacturing*, R. Roy, T. Furuhashi, and P. K. Chawdhry, Eds. London, U.K.: Springer-Verlag, 1999, pp. 521–535.

[20] H. Mühlenbein, T. Mahnig, and A. Rodriguez, "Schemata, Distributions and Graphical Models in Evolutionary Optimization," *J. Heuristics*, vol. 5, pp. 215–247, 1999.

[21] G. Harik, *Linkage Learning Via Probabilistic Modeling in the ECGA*. Berlin, Germany: Springer, 1999.

[22] M. Pelikan, D. E. Goldberg, and E. E. Cantú-paz, "Linkage problem, distribution estimation, and bayesian networks," *Evol. Comput*, vol. 8, no. 3, pp. 311–340, 2000.

[23] A. Johnson and J. L. Shapiro, "The importance of selection mechanisms in distribution estimation algorithms," in *Proc. 5th Int. Conf. Artif. Evol. AE01*, Oct. 2001, pp. 91–103.

[24] O. Cordón, I. F. de Viana, F. Herrera, and L. Moreno, M. Dorigo, M. Middendoff, and T. Stützle, Eds., "A new ACO model integrating evolutionary computation concepts: The best-worst ant system," in *Proc. Second Int. Workshop Ant Algorithms (ANTS'2000) From Ant Colonies to Artificial Ants*, Brussels, Belgium, 2000, pp. 22–29.

[25] N. Monmarché, E. Ramat, G. Dromel, M. Slimane, and G. Venturini, On the similarities between AS, BSC and PBIL: Toward the birth of a new meta-heuristic [Online]. Available: citeseer.ist.psu.edu/557467.html

[26] M. Dorigo, V. Maniezzo, and A. Colorni, "The ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst., Man, Cybern. Part B: Cybern.*, vol. 26, no. 1, pp. 29–41, Feb. 1996.

[27] Q. Zhang, J. Sun, E. Tsang, and J. A. Ford, "Estimation of distribution algorithm based on mixture: Preliminary experimental results," in *Proc. UKCI-02*, 2002, pp. 251–257.

[28] C. W. Ahn, K. P. Kim, and R. S. Ramakrishna, "A memory-efficient elitist genetic algorithm," in *Proc. 5th Int. Conf. Parallel Process. Appl. Math., PPAM'03*, 2003, pp. 552–559.

[29] C. W. Ahn, D. E. Goldberg, and R. S. Ramakrishna, "Multiple-deme parallel estimation of distribution algorithms: Basic framework and application," in *Proc. PPAM*, 2003, pp. 544–551.

[30] L. delaOssa, J. A. Gámez, and J. M. Puerta, "Initial approaches to the application of islands-based parallel edas in continuous domains," *J. Parallel Distrib. Comput.*, vol. 66, no. 8, pp. 991–1001, 2006.

[31] J. Madera, E. Alba, and A. Ochoa, *Towards a New Evolutionary Computation. Advances in the Estimation of Distribution Algorithms*, ser. Studies in Fuzziness and Soft Computing. New York: Springer, 2006, vol. 192, ch. A parallel island model for estimation of distribution algorithms, pp. 159–186.

[32] S. Zhou and Z. Sun, "Can ensemble method convert a 'weak' evolutionary algorithm to a 'strong' one?," in *Proc. Int. Conf. Comput. Intell. Modelling, Control, Autom. and Int. Conf. Intell. Agents, Web Technologies and Internet Commerce Vol-2 (CIMCA-IAWTIC'06)CIMCA'05*, Washington, DC, 2005, pp. 68–74.

[33] C. Gonzalez, J. Lozano, and P. Larranaga, "Analyzing the pbil algorithm by means of discrete dynamical systems," *Complex Syst.*, vol. 12, pp. 465–479, 2000.
[34] Q. Zhang, "On stability of fixed points of limit models of univariate marginal distribution algorithm and factorized distribution algorithm," *IEEE Trans. Evol. Comput.*, vol. 8, no. 1, pp. 80–93, Feb. 2004.
[35] Q. Zhang, J. Sun, and E. Tsang, "An evolutionary algorithm with guided mutation for the maximum clique problem," *IEEE Trans. Evol. Comput.*, vol. 9, no. 2, pp. 192–201, Apr. 2005.
[36] T. Hey, "Quantum computing: An introduction," *Comput. Control Eng. J.*, vol. 10, pp. 105–112, 1999.
[37] M. D. Platel, S. Schliebs, and N. Kasabov, "A versatile quantum-inspired evolutionary algorithm," in *Proc. IEEE Congr. Evol. Comput. CEC'07*, 2007, pp. 423–430.
[38] K.-H. Han and J.-H. Kim, "Quantum-inspired evolutionary algorithms with a new termination criterion, h_ϵ gate, and two phase scheme," *IEEE Trans. Evol. Comput.*, vol. 8, no. 2, pp. 156–169, Apr. 2004.
[39] A. A. da Cruz, M. Vellasco, and M. Pacheco, "Quantum-inspired evolutionary algorithm for numerical optimization," in *Proc. IEEE Congr. Evol. Comput. CEC 2006.* , Jul. 2006, pp. 2630–2637.
[40] H. Talbi, A. Draa, and M. Batouche, "A novel quantum-inspired evaluation algorithm for multi-source affine image registration," *Int. Arab J. Inf. Technol.*, vol. 3, no. 1, pp. 9–15, 2006.
[41] K. Kima, J. Hwang, K.-H. Han, J.-H. Kim, and K. Park, "A quantum-inspired evolutionary computing algorithm for disk allocation method," *IEICE Trans. Inf. Syst.*, vol. E86-D, no. 3, pp. 645–649, Mar. 2003.
[42] J. Jang, K.-H. Han, and J.-H. Kim, I. Press, Ed., "Face detection using quantum-inspired evolutionary algorithm," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2004, pp. 2100–2106.
[43] G. K. Venayagamoorthy and G. Singhal, "Quantum-inspired evolutionary algorithms and binary particle swarm optimization for training mlp and srn neural networks," *J. Comput. Theoretical Nanosci.*, vol. 2, no. 4, pp. 561–568, Dec. 2005.
[44] F. Liu, S. Li, M. Liang, and L. Hu, "Wideband signal doa estimation based on modified quantum genetic algorithm," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. 89, pp. 648–653, 2006.
[45] W. Zhou, C. Zhou, Y. Huang, and Y. Wang, *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*. New York: Springer, 2005, ch. Analysis of Gene Expression Data: Application of Quantum-Inspired Evolutionary Algorithm to Minimum Sum-of-Squares Clustering, pp. 383–391.
[46] J. L. Shapiro, "Drift and scaling in estimation of distribution algorithms," *Evol. Comput*, vol. 13, no. 1, pp. 99–123, 2005.
[47] H. Mühlenbein, "The equation for response to selection and its use for prediction," *Evol. Comput.*, vol. 5, no. 3, pp. 303–346, 1997.
[48] Q. Zhang and H. Mühlenbein, "On the convergence of a class of estimation of distribution algorithms," *IEEE Trans. Evol. Comput.*, vol. 8, no. 2, pp. 127–136, Apr. 2004.
[49] P. Larrañaga and J. A. Lozano, *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Boston, MA: Kluwer, 2002.
[50] C. W. Ahn and R. Ramakrishna, "Elitism-based compact genetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 7, no. 4, pp. 367–385, Aug. 2003.
[51] K. Sastry, D. E. Goldberg, and X. Llora, "Towards billion-bit optimization via a parallel estimation of distribution algorithm," in *Proc. 9th Annu. Conf. Genetic Evol. Comput. GECCO'07:* , New York, 2007, pp. 577–584.
[52] E. D. Weinberger, "NP completeness of Kauffman's NK Model, a tuneably rugged fitness landscape," Santa Fe Inst., 1996, Tech. Rep. 96-02-003.
[53] J. L. Shapiro, "Diversity loss in general estimation of distribution algorithms," in *Proc. PPSN*, 2006, pp. 92–101.
[54] D. E. Goldberg, *The Design of Innovation: Lessons From and for Competent Genetic Algorithms*. Norwell, MA: Kluwer, 2002.
[55] F. O. M. Wineberg, "Metrics for population comparisons in evolutionary computation systems," in *Intell. Syst. Control*, M. Hamza, Ed., Salzburg, Austria, 2003, vol. 388, pp. 24–24.
[56] M. Pelikan, K. Sastry, M. V. Butz, and D. E. Goldberg, "Hierarchical boa on random decomposable problems," in *Proc. 8th Annu. Conf. Genetic Evol. Comput. GECCO'06*, New York, 2006, pp. 431–432.

Michaël Defoin Platel received the M.Sc. degree in bio-informatics from the University of Paris 5, Paris, France, in 1999 and the Ph.D. degree in artificial intelligence from the University of Nice, Nice, France, in 2004.

From 2004 to 2006, he was funded by the CNES (the French National Center for Spatial Studies) to do postdoctoral research on the ocean color inverse problem. In 2007, he moved to New Zealand to develop new estimation of distribution algorithms at the Knowledge Engineering and Discovery Research Institute (KEDRI). Since October 2007, He has been a member of the Computer Science Department, University of Auckland, Auckland, New Zealand. His main research activities are machine learning, evolutionary algorithms, inverse problems solving, and bio-informatics.

Stefan Schliebs received the M.Sc. degree in computer science from the University of Leipzig, Leipzig, Germany. He is currently pursuing the Ph.D. degree at the Knowledge Engineering and Discovery Research Institute (KEDRI), Auckland, New Zealand, under the supervision of Prof. N. Kasabov and Dr. M. D. Platel.

His research interests include novel evolving connectionist models inspired by the principles of quantum theory, artificial neural networks and computational intelligence, and evolutionary algorithms, especially in the area of feature subset selection.

Nikola Kasabov (SM'96) received the M.Sc. and Ph.D. degrees from the Technical University of Sofia, Sofia, Bulgaria, in 1971 and 1975, respectively.

He is the Founding Director and the Chief Scientist of the Knowledge Engineering and Discovery Research Institute (KEDRI), Auckland, New Zealand. He holds a Chair of Knowledge Engineering at the School of Computing and Mathematical Sciences, Auckland University of Technology. His main research interests are in the areas of intelligent information systems, soft computing, neurocomputing, bioinformatics, brain study, speech and image processing, novel methods for data mining, and knowledge discovery. He has published more than 400 publications that include 15 books, 120 journal papers, 60 book chapters, 32 patents, and numerous conference papers. He has extensive academic experience at various academic and research organizations: University of Otago, New Zealand; University of Essex, U.K.; University of Trento, Italy; Technical University of Sofia, Bulgaria; University of California at Berkeley; RIKEN and KIT, Japan; Technical University Kaiserslautern, Germany, and others.

Dr. Kasabov is a Fellow of the Royal Society of New Zealand and a Fellow of the New Zealand Computer Society. He is the President-Elect of the International Neural Network Society (INNS) and the President of the Asia–Pacific Neural Network Assembly (APNNA). He is a member of several technical committees of the IEEE Computational Intelligence Society and of the IFIP AI TC12. He is on the editorial boards of several international journals, including the IEEE TRANSACTIONS ON NEURAL NETWORKS, the IEEE TRANSACTIONS ON FUZZY SYSTEMS, *Neural Networks*, *Information Science*, and the *Journal of Theoretical and Computational Nanoscience*. He chairs a series of international conferences ANNES/NCEI in New Zealand.