Toward a Framework for Quantum Evolutionary Computation

Donald A. Sofge
Navy Center for Applied Research in Artificial Intelligence
Naval Research Laboratory
Washington, DC USA

***Abstract*— Biological evolution provides an immensely powerful toolset for problem solving, while evolutionary computation attempts to harness the power of biological evolution for solving problems using classical computing paradigms. Quantum computing offers many apparent advantages over classical computing for certain types of problems, such as searching or optimizing over large solution sets. Once practical quantum computers are available, we would like to take advantage of their highly parallel computing capabilities for use in evolutionary computation. In this work we explore the nexus between quantum and evolutionary computation, and propose an approach toward a practical framework for performing evolutionary computation on quantum computers.**

***Keywords*—evolutionary computation, quantum computing, biological evolution**

I. INTRODUCTION

THE availability of practical quantum computers may have a significant impact on the use of evolutionary computing techniques. Quantum computing offers the potential to fundamentally change evolutionary computation as we know it by allowing certain operations, which on a classical computer require exponential computational complexity, to be performed in polynomial time. The exploration of certain state spaces may be performed in parallel on a quantum computer such that all possible states are explored simultaneously.

Pure quantum computation, however, will likely be of limited value. Only through integration of hybrid quantum and classical computing hardware will significant advances be achieved in rendering currently intractable problems (under classical computing) tractable. One line of research for integrating quantum and evolutionary computation focuses on the use of genetic programming techniques to automatically generate quantum programs [1]. In this work we consider a different tack: that the availability of quantum computing hardware will allow new opportunities for evolutionary computation.

A significant worldwide effort is underway to build the first practical (non-trivial) quantum computer (that is, a quantum computer that can solve a problem not more easily and trivially solved using a classical computer). To date quantum computers have been built containing a small number of qubits (fewer than a dozen). Producing a working quantum computer is a daunting challenge, involving researchers in a variety of disciplines ranging from physics to chemistry to material science to mathematics to computer science. Nonetheless, several physicists and other practitioners of science argue that there is significant evidence that biological systems already use quantum computing as part of the suite of biological processes that give rise to and sustain life, as described below.

Since practical quantum computers are not yet available, and only very crude and simple quantum computers have been simulated on classical computing platforms, the scope of this effort is limited to considering the potential for performing evolutionary computation on a quantum computer, and addressing some of the obstacles to achieving this goal.

A. *Biology, Evolution, and Computation*

Biological evolution and quantum evolution both provide mechanisms for change and adaptation of natural systems. In biological evolution the processes involved operate at the macroscopic level and principally concern complex systems of organic molecules which have self-organized to possess the property we call "life". Such systems perform an astonishing array of complex activities which allow them to grow, eat, breathe, reproduce, and pass their genes on to subsequent generations, thereby preserving some semblance of their genetic identity into the future, well beyond the lifespans of the individual organisms.

Quantum evolution (the meaning of the term *evolution* as used here is distinguished from its meaning as used in biological evolution), on the other hand, describes change and adaptation of natural systems at the very smallest scales, such as the interactions between subatomic particles, photons of light, and in some cases, atoms and molecules. Modern theories of quantum mechanics describe particles as existing as wave functions in superposed or entangled states such that the probability of measuring the system in one state or another depends upon the observation being made and whatever interactions may occur with the particle(s) and the outside environment.

This work was supported by the Naval Research Laboratory under NRL Work Order N0001406WX30002.

Whereas evolutionary computation (EC) seeks to apply techniques inspired by natural biological evolution to solve challenging practical problems in engineering, the sciences, and elsewhere, quantum computation is the application of quantum mechanical phenomena to solve computational problems. Quantum computing offers the potential to solve a number of computational problems, such as the factoring of a large number into the product of primes for breaking complex cryptographic systems, in polynomial time. Such problems may be solved using classical computing techniques only with exponentially increasing amounts of computing time and/or memory. Whereas these problems become computationally intractable under classical computing as the problem size increases (with problem size indicated by some problem dependent parameter such as the number of digits of a large composite number), under quantum computing the problem may still be tractable, increasing in complexity as only a polynomial function of the problem size.

We begin our analysis of the intersection of evolutionary and quantum computation with a review of some of the more prominent claims regarding the use of quantum effects in living systems, and in particular optimization using biological quantum computers.

II. Biological Quantum Computation

The role of quantum computing in biological evolution is currently the subject of active investigation. It is plausible that quantum phenomena, if not quantum computing, play a role in certain DNA synthesis and cellular processes, and may in fact play a crucial and necessary role which cannot be achieved using classical mechanisms. If the means for achieving quantum mechanisms using biochemical molecules at the cellular level are available to biological evolution in order to improve upon non-quantum optimization effects, it would seem reasonable that evolution would in fact take advantage of these quantum mechanisms. In this section we look at evidence and argument by a range of scientists on the role of quantum computing on biological evolution.

A. Patel's Quantum Genetic Synthesis Model

Apoorva Patel, a physicist at the Indian Institute of Science in Bangalore, argues that the human body, and every other living thing, is teeming with quantum computers doing everything from organizing protein structures to copying DNA to regulating cellular processes [2]. While it is well known and widely accepted that quantum mechanics plays an important role in organic chemistry, with tautomeric bonds, potential wells, quantum tunneling effects and the like, little attention has been given to the existence of macroscale effects of quantum mechanics. Patel argues that nature used the computational power of evolution in order to design microscopic quantum computers billions of years ago, and that these quantum computers are an essential ingredient for life. Patel begins his argument by describing life (in particular, biomolecular life) as a form of computation. Computation is the processing of information. Biological structures such as DNA and RNA are then simply means for encoding information. DNA and RNA are strings composed from an alphabet of 4 letters (hence, a quaternary system as opposed to binary), and proteins are composed from an alphabet of 20 amino acids.

One of the key mysteries of biology Patel seeks to answer is why nature evolved a 4-character (quaternary) alphabet for representing DNA rather than a 2-character binary one. Intuition would suggest that binary would be simpler, more efficient, and less prone to errors. Patel argues that using quantum computing, the 4-character alphabet is in fact more efficient because certain matching operations of DNA, such as connecting the appropriate base pairs together, is actually more efficient than would be the case with a binary alphabet. The matching of base pairs actually involves a superposition of possible matches, with quantum interference steering the bases to their proper complements in the chain.

The superiority of quantum computing over classical computing for searching databases (in this case a database of 4 types of entries) has been proven using Grover's algorithm (Section III.B below). In the case of DNA replication, the speedup provided by quantum computing is a factor of two. Patel writes *"... it is imperative to investigate whether DNA has the quantum hardware necessary to implement the quantum search algorithm*."

B. McFadden's Quantum Adaptive Mutation Model

Standard theories of Darwinian evolution hold that mutations occur randomly, and that the direction of evolutionary change is determined by selection mechanisms imposed by the environment to favor more highly fit individuals. Recent studies of the frequency of mutations in bacteria and eukaryotes dispute this notion of mutations occurring randomly, and instead point to the possibility of an adaptive mutation mechanism. Adaptive mutations differ from standard random mutations in that they (i) only occur in cells that are either not dividing or are dividing only rarely, (ii) are time-dependent, not replication dependent, and (iii) appear only after the cell has been exposed to the selection pressure. Johnjoe McFadden [3] argues that quantum mechanical effects accelerate the rate of mutation for genomes through interference of the wave function of the genome with the environment, thereby providing a mechanism for adaptive mutation. McFadden further argues that spontaneous mutations may be initiated by quantum events such as the tunneling of a single proton from one site to another. He justifies his argument by studying the decoherence times of protons within DNA, and shows that based upon such calculations, DNA strings may maintain coherence over the time scales required for mutations to occur.

The significance of quantum mechanics in initiating mutations has been explored by many researchers since even before the discovery of the structure of DNA, including Delbuck *et al.* [4], Schrödinger [5], and Watson and Crick [6]. Recent work by Goswami and Todd [7] and Ogryzko [8] describe DNA as existing in a superposition of mutational states, with the wave function collapsing due to interaction of the cell with the environment. McFadden [3] argues that the mechanism of interaction that leads to the favorable mutation is a dense series of “measurements” which forces the quantum system to evolve along a desired path. He compares this to the inverse quantum Zeno effect described by Aharonov and Vardi [9] and Altenmuller and Schenzle [10]. McFadden suggests that “*living cells could act as biological quantum computers, able to explore multiple possible mutational states and collapse towards those states that provide the greatest advantage*.” Proving that living cells can and do exist in superposed quantum mutational states, however, will require a better understanding of quantum mechanical effects on macroscopic biological systems.

C. Penrose’s Quantum Consciousness

The British mathematician Roger Penrose is largely responsible for popularizing the connection between quantum phenomena and consciousness. In his 1989 book *The Emperor’s New Mind* [11] and his 1994 book *Shadows of the Mind* [12] Penrose argues that based upon Gödel’s theorem, human thought is non-algorithmic, and not computable. He argues that quantum mechanical phenomena lie at the heart of free will, and that free will requires consciousness. He points out that there is no current understanding or theory of the collapse of the wave function, and proposes that a theory of quantum gravity would fill the gap. He further argues that no classical computer will ever achieve consciousness (or intelligence) because it doesn’t have free will, and goes on to argue that the seat of consciousness in the brain is the microtubules which form the cytoskeletons of neurons. Penrose speculates that microtubules play a key role in neural functioning, and serve to interact with the quantum gravity effect to achieve non-algorithmic computing.

Many of Penrose’s assertions seem quite speculative in nature due to the lack of supporting scientific evidence establishing the role of quantum macroscale effects in biological systems. However, Penrose does help to elucidate where current theories of the effects of quantum phenomena on biological systems fall short.

D. De Garis’ Quantum Brains

Hugo de Garis [13] claims that when quantum computers are finally built, they will quickly make the field of evolutionary computing obsolete. The heart of de Garis’ argument is that evolutionary computation is merely sampling a solution space of 2^N individuals (where N represents the number of bits in the genome). A quantum computer with N qubits will be able to search the entire space simultaneously by representing the genome as a superposition of qubits. De Garis speculates that just as Moore’s Law doubles the number of transistors on a computer chip every eighteen months, improvements in quantum computers will increase the number of qubits at an exponential rate until eventually it exceeds Avogadro’s number (6.022×10^{23}), at which point “*it will be difficult to imagine any problem that will not succumb to its power*.” Despite the lack of relevance to the central thesis and conclusions of the paper, most of the work described therein is concerned with replacing evolutionary computing with a quantum computing framework to design and train neural networks. De Garis refers to neural networks derived in this manner, and run on a quantum computer, as *Quantum Artificial Brains*.

De Garis’s arguments are far from convincing for a variety of reasons. First, the challenge of scaling quantum computers to include larger numbers of qubits is completely ignored. Maintaining coherence of superposed and entangled quantum states, and structuring quantum interference effects to achieve some desired computational goal, for large quantum systems are tremendously difficult unsolved problems.

Second, simply superposing input values in qubits doesn’t solve the problem of doing a massively parallel search through a complex solution space. The application of EC techniques to complex real-world problems often requires computer runs lasting days, weeks, or even months, even on a highly parallel computing platform. The reason for the long run times is often not the overhead due to the evolutionary algorithm, or even necessarily the size of the solution space; rather, it is usually the time-consuming evaluation of each individual.

Suppose we want to evolve a flight control system that consists of 50 binary parameters. Even if we could assign a qubit to represent each parameter, the computationally intensive part of the evolutionary process is running the simulation for each genome case to generate a fitness value. A quantum computer designed to simultaneously evaluate every possible flight scenario to optimize a parameter set, the flight control laws, could in theory do this. However, the challenge of the task is then shifted to the design of such a quantum computer. Merely having a 50 qubit register for the inputs to the quantum computer doesn’t tell us how to build a quantum flight simulator.

De Garis does point out the need for quantum compilers to help design software and circuits for quantum computing, although his motivation seems largely geared to the generation of simulations of “*quantum-neural-network-based artificial quantum brains*.”

The role of quantum computing in biological evolution is currently the subject of much conjecture, speculation, argumentation, bold proclamation, and perhaps some bits of solid evidence. It seems plausible that quantum phenomena, if not quantum computing, play a role in certain DNA synthesis and cellular processes, and may in fact play a crucial and necessary role which cannot be achieved using classical mechanisms. If the means for achieving quantum mechanisms using biochemical molecules at the cellular level were available to biological evolution in order to improve upon non-quantum optimization effects, it would seem reasonable that evolution would in fact take advantage of these quantum mechanisms.

In the next section we turn our attention to the specific advantages offered by known quantum algorithms over classical techniques, and address how we might incorporate these into a practical framework for quantum evolutionary computation (QEC), which we define as evolutionary computation performed on a quantum computer.

III. Foundations for a Practical Framework for Quantum Evolutionary Computation

A. Deutsch-Jozsa Algorithm

The most dramatic advantage in computational speed-up of quantum computation over classical computation is perhaps best shown with the Deutsch-Jozsa algorithm [14]. The purpose of the algorithm is:

Given a number x such that $0 \le x \le n-1$, and
Given a function $f(x)$ guaranteed to be either constant or balanced for all values of x
Calculate function $f(x)$ such that either:
$f(x)$ is constant for all x (either all zeros or all ones), or
$f(x)$ is balanced for all x (half zeros and half ones).
Return 0 if $f(x)$ is constant, and 1 if $f(x)$ is balanced.

While a classical algorithm requires $2^n/2+1$ evaluations to perform the calculation, a quantum computer using the Deutsch-Jozsa algorithm can perform the calculation using a single evaluation.

The challenge of formulating a framework for QEC based upon the Deutch-Jozsa algorithm is that the fitness evaluation requirement for evolutionary computation would need to be mapped onto the constant/balanced property of f, a non-trivial task. In general, while the speed-up achieved by Deutsch-Jozsa is truly impressive, to date there are no known applications of this algorithm to solve real computational problems. A computational problem that shows up much more frequently is search of a large database to find an entry or entries possessing certain features or qualities.

B. Grover's Algorithm

In 1995 Lov Grover [15] showed that a quantum computer could search an unstructured solution space faster than a classical computer could. Where the space to be searched consists of N unstructured entries, a classical algorithm requires O(N) operations, whereas a quantum computer running Grover's algorithm could do it in O($\sqrt{N}$) operations. While this is not as dramatic a speed-up as offered by the Deutsch-Jozsa algorithm, or even the speedup offered by Shor's algorithm for factoring large composite numbers, it is still significant.

Suppose we want to optimize a solution to a problem that has a solution space of size 2^n, where n is the number of bits in the genome. Doing a complete search of the space using a classical algorithm would require O(2^n) operations. A complete search using Grover's algorithm would require O($2^{n/2}$) operations, a substantial savings.

However, we must ask if we can do better. After all, the point of the evolutionary algorithm is not to perform a thorough search of a completely unstructured solution space, but rather to sample a well-structured solution space such that each generation our sampling moves us closer to some extrema points in the space that represent better solutions. What we need is a variant of Grover's algorithm that instead of exhaustively searching the solution space instead seeks to minimize or maximize some value or property over the space of solutions. This is exactly what we find with Dürr's algorithm.

C. Dürr's Algorithm

Given an unsorted list of N items, Dürr's algorithm [16] finds the index of the item with minimum value with probability of at least 1/2. The algorithm requires O($\sqrt{N}$) probes of the list. This requires O(N) operations on a classical computer. If the list is represented as a table T[0..N-1], then the minimum search problem is to find the index y such that T[y] is minimized. The algorithm utilizes the quantum exponential search algorithm given in [17], and proceeds as follows.

Quantum Minimum Searching Algorithm [16]:

1. Let $c \ge 1$ and m_0 be given
   Choose threshold index y uniformly at random from $\{0, \ldots, N-1\}$
2. Repeat the following until the total number of iterations is more than $c\sqrt{N}$
   a. Initialize $|\Psi_0\rangle = \sum_i \frac{1}{\sqrt{N}} |i\rangle|y\rangle$
      Mark each item i for which T[i]<T[y]
   b. Apply quantum exponential search algorithm [17] to the first register, with a timeout of $2m_0$ iterations
   c. Observe the first register: let y' be the outcome
      If T[y']<T[y], then set threshold index y to y'
3. Return y

As with many quantum algorithms, the value returned has a certain probability of being incorrect. The algorithm may be rerun a number of times to reduce this probability of incorrectness to whatever level is desired.

Dürr's algorithm points us in a useful direction for establishing a practical framework for QEC. Notice that the evaluation component, which we have previously argued is extremely important, is a simple comparison of two values. This is overly simplistic for most computational challenges to which we might apply an evolutionary algorithm. However, if we have our eval function (such as the flight simulator discussed previously) implemented on a quantum computer, then Dürr's algorithm shows us how we might find the minimum (or it could just as easily find the maximum). No evolutionary algorithm would be needed, since we are in fact searching the entire space.

D. Not All Solution Spaces are Unstructured

Up to this point we have focused upon quantum algorithms that search through completely unstructured solution spaces. As the *no free lunch theorem* [18] tells us, no one algorithm is better than any other if no structure is imposed. Machine learning techniques that try to generalize and "learn" classifications or mappings across state-spaces inherently depend upon the existence of some sort of exploitable structure.

Evolutionary algorithms and other beam-search methods use gradients as well as highly-disruptive crossover operators to escape from local minima, and rely upon a population of individuals to explore the solution space. Other techniques are very efficient at searching tree structures. Whatever framework is used to achieve QEC, it will inevitably require that the algorithm exploit whatever structure may be present in order to improve performance of the search. The worst-case-scenario is that if no structure may be exploited, then we resort to Dürr's algorithm and expect no better than O($\sqrt{N}$) performance.

E. Sampling versus Searching the Solution Space

Another key difference between quantum search algorithms given to date and evolutionary computation algorithms is that the quantum search algorithms search the entire solution space rather than selectively sampling it. We can certainly sample the solution space with our quantum computers as well, but why would we?

Some possible reasons might include that the quantum computer doesn't possess a sufficient number of qubits to process all possible states of the eval function simultaneously, that not all of the available qubits can be mutually entangled, or that quantum gates only entangle a few qubits at a time. Suppose that our quantum computer requires 1000 qubits, but our state-of-the-art quantum computer may only entangle 10 qubits at a time (more than is currently feasible), and that our quantum compiler is set up to optimize the quantum code to use the maximum number of qubits available. The quantum computer is able to run many parts of the simulation simultaneously in parallel, but because it can't entangle enough qubits simultaneously (due to quantum gate fan-in fan-out limitations), it must store the results of intermediate states, possibly resulting in decoherence, and then reload the quantum gates with other pieces of code to continue. Thus, the quantum computer uses a stored program just as does a von Neumann classical computer, but in this case the stored program is a quantum algorithm rather than a classical one. We might wish for a quantum mass storage device on which it would be possible to write intermediate results without having them decohere. However, the no-cloning theorem [19] tells us that it is impossible to exactly copy a quantum state without having it decohere. Thus, it would seem that quantum mass storage is out. A small glimmer of hope may exist, however, with the term *exactly*. It may be possible to make a copy of a quantum state and maintain some level of coherence while also accepting some level of error or *inexactness* of the copy.

By sampling rather than searching the solution space, the quantum search algorithm is able to utilize the dynamics of evolution to move the solution set toward a more global optimum. The speedup achieved by sampling over searching in the quantum computing framework would depend heavily upon the problem space and the efficiency of the evolutionary algorithm employed. This is clearly a function of how well the evolutionary algorithm employed performs on the problem space.

F. Parallel Distributed Evolutionary Algorithms

While most of the discussion up to this point has assumed use of sequential generational evolutionary algorithms (EAs), an attractive alternative is the use of parallel distributed EAs [20]. Parallel distributed EAs are designed to run more efficiently on parallel hardware, but may be used advantageously on sequential processors as well [21]. Unlike sequential generational EAs that require global knowledge of the population in order to achieve progress, parallel distributed EAs may require no global knowledge. Instead, populations are represented using island models or diffusion grids [22] such that local sub-populations coevolve, and individuals periodically migrate to (or genetic information is otherwise exchanged with) neighboring groups.

Parallel distributed EAs may have significant advantages over sequential generational EAs in the QEC framework. Elimination of fixed generational boundaries, and with no global knowledge of the population assumed, may allow quantum processors modeling individuals to become entangled, perform crossover and mutation (e.g. through quantum gate operations), and produce offspring while maintaining a coherent state. A similar effect may be possible using binary tournament selection with a generational EA; however, the overhead of running the tournament, sorting, truncation, etc. would still most likely be handled by a classical computer.

G. Toward a QEC Framework

The preceding sections address several points to consider in formulating a practical framework for QEC. First, quantum computing may play a significant role in biological evolution from which we could draw inspiration for design of QEC algorithms. However, our current state of knowledge in this area is woefully lacking. First we should establish that there indeed are quantum phenomena operating on the biological scale that somehow affect evolution.

One of the most readily measurable properties of biological systems that may shed light on whether or not quantum phenomena are operating at a macroscopic scale is adaptive mutation rate. Through efficient DNA sequence analysis we can isolate the mutation rates at specific loci on the genomes of microbes. It may be possible to design quantum genetic experiments on a large number of microbes with a strong tendency to mutate at these loci by controlling external conditions which are more (or less) favorable to certain mutations, allowing the microbes to mutate in an unobserved state for some number of generations while applying (or not) the external stimulus (in a double-blind procedure), genetically sequencing the offspring and observing the resulting mutation rates. Statistical analysis should provide insight into the expected values for favorable mutations at the chosen loci assuming a random (but constant) rate of mutation, versus the differing result one might expect from a quantum adaptive mutation rate as predicted by [3]. Of course, merely showing that a mutation rate is adaptive doesn't prove that quantum mechanics has anything to do with it. If there is a quantum causal effect, it should be detectable based upon the observations of the experimenters.

Quantum algorithms for searching unstructured sets have been shown to be superior to their classical equivalents [15], but for QEC we are more interested in searching structured sets (solution spaces). Dürr's algorithm [16] provides one approach to finding the index of the least item in an unstructured list. Quantum algorithms for searching trees, directed graphs, linked lists, and surfaces with gradients would provide more secure footing for implementing evolutionary computation algorithms on a quantum computer.

Finally, the type of evolutionary algorithm employed may play an important role in implementing it under a QEC framework. Parallel distributed EAs offer a number of intrinsic characteristics better suited to quantum parallelism than sequential generational EAs. Representation of the genomes is critical, as is implementation of genetic operators and eval functions through quantum gate designs (hardware implementations of quantum algorithms). It is conceptually feasible to entangle k qubits in a single quantum register, and have the contents of that register represent all possible genotypes (or at least sub-genotypes of length k). A potentially more daunting task would be to design a quantum super-gate that takes these qubits as inputs and simultaneously evaluates them, returning the "fittest" individual, or perhaps returning fitness scores for all (in a superposed state). The key is that if we can avoid falling into a decoherent state as much as possible, the efficiency of the QEC method over standard (classical computing) EC approaches will be maximized. If we can represent the entire genome population in a quantum register and perform all possible evals simultaneously, then there is no need to run an EA; we need only perform an exhaustive search of the problem space to find the optimum. If this is not possible (and it is the author's opinion that it will not be possible except for trivial problems), then we need to figure out to decompose the problem for optimization using an EA in the QEC framework.

If the key restriction on the implementation of practical quantum processors is the number of qubits that may be simultaneously entangled in a quantum register, or that may be maintained in a coherent entangled state through a quantum gate transform, then clearly we need to design parallel distributed EAs to take advantage of the available architecture. The key unknown for future research to resolve is how much and what types of information may be entangled (genes? genomes? base-pairs?), how may this information be searched using quantum algorithms (and their physical instantiation as quantum gates), and how may we store intermediate results from quantum operations. QEC solutions will most likely involve integration of both quantum and classical computing components, with algorithms tailored to take advantages of the strengths of each.

IV. Summary and Conclusions

This paper explores issues in quantum computing and evolutionary computation relevant to creation of a practical framework for quantum evolutionary computation whereby evolutionary computation may be performed on a quantum computer. In so doing we explore current evidence, theory and scientific conjecture regarding the role of quantum computing in biological evolution, we review key algorithmic developments of quantum computing as they might apply to QEC. We look at approaches to evolutionary computation itself, and discuss how one form of evolutionary computation may be better suited to implementation in a QEC framework than another. Finally, we provide the broad outlines of a practical framework for QEC, and discuss how developments in quantum and evolutionary computation may help move us toward integration of the two.

References

[1] L. Spector, *Automatic Quantum Computer Programming: A Genetic Programming Approach*, New York: Springer-Verlag, 2004.

[2] A. Patel, "Quantum Algorithms and the Genetic Code," *Proc. Winter Institute on Foundations of Quantum Theory and Quantum Optics*, 2001.

[3] J. McFadden, J. Al-Khalili, "A quantum mechanical model of adaptive mutation," *Biosystems* 50, 1999, pp. 203-211.

[4] M. Delbruck, N. W. Timofeeff, K. G. Zimmer, *Nachr. Biol. Ges. Wiss. Gottingen 1*, 189, 1935.

[5] E. Schrödinger, E., "What is Life?" London: Cambridge University Press, 1944.

[6] J. D. Watson, F. H. C. Crick, "Genetic implications of the structure of deoxyribonucleic acid," *Nature* 171, 1953, pp. 964-969.

[7] A. Goswami, D. Todd, "Is there conscious choice in directed mutation, phenocopies, and related phenomena? An answer based on quantum measurement theory." *Integr. Physiol. Behav. Sci.* 32, 1997, pp. 132-142.

[8] V. V. Ogryzko, "A quantum theoretical approach to the phenomenon of directed mutations in bacteria (hypothesis)," *Biosystems* 43, 1997, pp. 83-95.

[9] Y. Aharonov, M. Vardi, "Meaning of an Individual Feynman Path," *Phys. Rev. D* 21, 1980, pp. 2235-2240.

[10] T. P. Altenmuller, A. Schenzle, "Dynamics by measurement: Aharonov's inverse quantum Zeno effect," *Phys. Rev. A* 48, 1993, pp. 70-79.

[11] R. Penrose, *The Emperor's New Mind*, Oxford University Press, 1989.

[12] R. Penrose, *Shadow's of the Mind*, Oxford University Press, 1994.

[13] H. de Garis, Z. Zhang, "Quantum Computation vs. Evolutionary Computation: Could Evolutionary Computation Become Obsolete?" *Congress on Evolutionary Computation (CEC2003)*, 2003.

[14] M. A. Nielsen, I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2000, pp. 34-38.

[15] L. K. Grover, "A fast quantum mechanical algorithm for database search," *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, 1996, pp. 212-219.

[16] C. Dürr, P. Høyer, "A quantum algorithm for finding the minimum," *e-print quant-ph/9607014,* 1996.

[17] M. Boyer, G. Brassard, P. Høyer, A. Tapp, "Tight bounds on quantum searching," *Fortschritte Der Physik*, 1998.

[18] D. H. Wolpert, W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, 1997, pp. 67-82.

[19] W. K. Wootters, W. H. Zurek, "A single quantum cannot be cloned", *Nature*, **299**: 802-803, 1982.

[20] E. Alba, J. M. Troya, "A survey of parallel distributed genetic algorithms," *Complexity*, Vol. 4, Issue 4, 1999, pp. 31-52.

[21] V. S. Gordon, D. Whitley, "Serial and parallel genetic algorithms as function optimizers," *Proceedings of the 5th ICGA*, S. Forrest (Ed.), Morgan Kaufmann, 1993.

[22] P. Spiessens, B. Manderick, "A massively parallel genetic algorithm," *Proceedings of the 4th ICGA*, R .K. Belew and L. B. Booker (Eds.), Morgan Kaufmann, 1991.