

A Quantum-Inspired Differential Evolution Algorithm for Solving the N-Queens Problem

Amer Draa¹, Souham Meshoul², Hichem Talbi³, and Mohamed Batouche²

¹Computer Science Department, Mentouri University, Algeria

²Computer Science Department, King Saud University, Saudi Arabia

³OE Faculty, Emir Abdelkader University, Algeria

Abstract: *In this paper, a quantum-inspired differential evolution algorithm for solving the N-queens problem is presented. The N-queens problem aims at placing N queens on an $N \times N$ chessboard, in such a way that no queen could capture any of the others. The proposed algorithm is a novel hybridization between differential evolution algorithms and quantum computing principles. Accordingly, differential evolution algorithms have been enhanced by the adoption of some quantum concepts such as quantum bits and states superposition. The use of the quantum interference has allowed this hybrid approach to have a remarkable efficiency and good results.*

Keywords: *Quantum computing, differential evolution, N-queens problem, combinatorial optimization.*

Received March 20, 2008; accepted July 8, 2008

1. Introduction

Combinatorial optimization problems form a set of problems which need a considerable effort and time to be solved. Their difficulty lies in the fact that there is no formula for solving them exactly. Every possibility has to be examined in order to find the best solution and the number of possibilities increases exponentially as the size of the problem increases [3].

Classical methods are very limited in solving such problems because a large set of data is to be processed and a large solution space must be explored. So, other methods, known as approximate methods, are used to solve such problems. The aim of an approximate method is to get an accepted solution, not necessary the optimal one. Genetic algorithms, artificial neural networks, simulated annealing, taboo search and Differential Evolution Algorithms (DEA) are among the most famous approximate methods used in the combinatorial optimization field.

Differential evolution algorithms are population based algorithms like genetic algorithms using similar operators; crossover, mutation and selection. The main difference is that genetic algorithms give more importance to the crossover operation while DEAs consider the mutation operation as the most important one. The DEA's main operation is based on the differences of randomly sampled pairs of solutions in the population [6]. Like in evolution strategies, the differential evolution is oriented to optimize real values, but it is very simple to be implemented [8].

Another search field called Quantum Computing (QC) has appeared and induced intense researches in the last decade. This evolution that takes its origins

from the quantum physics principles reduces remarkably the algorithms' complexity. This is offered by the possibility of parallel computing. Such a possibility of parallel computing can be exploited to solve combinatorial optimization problems which use a great mass of data. Thus, quantum computing gives the opportunity of designing very powerful and efficient algorithms. However, these algorithms may not be well exploited before the construction of powerful quantum machines. Awaiting the construction of such machines, the idea of simulating quantum algorithms on classical computers or to combine them to other conventional methods has emerged [3].

Researches in the field of combination between evolutionary algorithms and quantum computing have started at the end of the 1990's. The purpose of this combination is to enhance the profit of each one of these two approaches by mutually inspiring each from the other. Within this sphere, we are interested in studying the differential evolution-quantum hybridization and its contribution in solving combinatorial optimization problems.

In this paper, we propose a quantum-inspired differential evolution algorithm for solving the N-Queens Problem (NQP). A previous instance of QDEAs has been applied for image registration [2] and it has given good results; that has motivated us to apply it on the NQP and to compare its performance to that of another work [3] that uses a conventional Quantum-inspired Genetic Algorithm (QGA) for solving the NQP.

Consequently, the rest of this paper will be organized as follows. Section 2 gives some concepts

about differential evolution, NQP and quantum computing. The proposed approach is described in section 3. Section 4 illustrates some experimental results and discussions. Finally, conclusion and future directions are drawn up.

2. Basic Concepts

2.1. N-Queens Problem

The NQP is a classical artificial intelligence problem. It is a general case of the 8-Queens problem. This combinatorial optimization problem has been studied for more than a century. The 8-Queens problem was first introduced by a chess player, Max Bezzel, in 1848. Since 1850, the problem has attracted the attention of several famous mathematicians including Gauss, Polya, and Lucas.

The N-Queens problem can be defined as follows: place N queens on an N x N chessboard, each queen on a square, so that no queen could capture any of the others, that is, a configuration in which there exists at most one queen on a given row, column or diagonal. A solution for the 8QP is given in Figure 1.

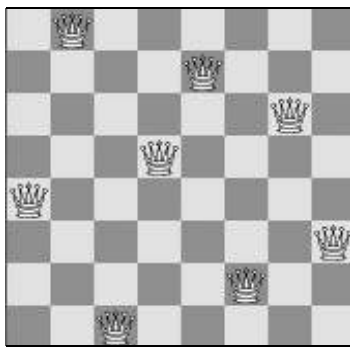


Figure 1. A possible solution of the N-queens problem where N=8.

During the last three decades, the problem has been discussed in the context of computer science and used as an example of backtrack algorithms, permutation generation, divide and conquer paradigm, program development methodology, constraint satisfaction problems, integer programming, specification and neural networks [1, 12].

A common way to solve this problem consists in trying to put the queens on the board squares one after the other. If one queen threatens the newly introduced queen, we withdraw the queen and search for another position. If we cannot find a solution, we choose to remove a queen already positioned, assign it another position that has not yet been used, and start the search again. This last operation is called a back-track, and the whole strategy is called a trial-and-error algorithm. It is known that for $N=8$, there are exactly 92 solutions, or less if we consider symmetric solutions as equal [1, 12].

The number of solutions for $n = 1, 2, \dots, 15$, is 1, 0, 0, 2, 10, 4, 40, 92, 352, 724, 2680, 14200, 73712, 365596, 2279184.

2.2. Differential Evolution

Differential Evolution (DE) is a novel parallel direct search method which uses N parameters vectors x_i, G , $i=0..N-1$ as a population for each generation G. N does not change during the minimization process. A Differential Evolution Algorithm (DEA) is a stochastic, population-based optimisation algorithm. It aims at optimizing real parameter and real valued functions [2, 8].

The crucial idea behind DE is a new scheme for generating trial parameter vectors. DE generates new parameter vectors by adding the weighted difference vector between two population members to a third member. If the resulting vector yields a lower objective function value than a predetermined population member, the newly generated vector replaces the vector with which it was compared. The comparison vector can, but does not need to be a part of the generation process mentioned above.

The main evolutionary operator in DE is meaningfully different from other evolutionary algorithms since mutation is neither based on the alteration of genes by using a mutation probability nor rest on the use of a defined probability distribution function. In DE the mutation operator mutates μ vectors through the weighted difference of two (or four) other vectors according to:

$$\vec{V}_i = \vec{a}_i + F * (\vec{a}_{r1} - \vec{a}_{r2}) \quad (1)$$

where $i = 1, 2, \dots, \mu$, and the random indexes $r1$ and $r2 \in [1, 2, \dots, \mu]$ are mutually different and also distinct from the index i . F is a real constant belonging to the interval $[0, 2]$ which affects the differential variation between two vectors. As in other evolutionary algorithms, the crossover operator is introduced in order to increase the diversity of the population. The general structure of a differential evolution algorithm is shown in Figure 2.

```

Generate random solutions that cover the given space.
Evaluate each solution.
g=1
While (convergence is not reached)
  For i=1 to Population size
    Apply differential mutation.
    Execute differential crossover
    Clip the solutions if necessary
    Evaluate the new solution.
    Apply differential selection.
  End
  g=g+1;
End

```

Figure 2. General structure of the DEA.

2.3. Quantum Computing

In early 80, Richard Feynman's observed that some quantum mechanical effects cannot be simulated efficiently on a computer. His observation led to speculation that computation in general could be done more efficiently if it used this quantum effects. This speculation proved justified in 1994 when Peter Shor described a polynomial time quantum algorithm for factoring numbers [9]. In this section the basic concepts of quantum computing such as quantum bit, states superposition, quantum measurement, and quantum gates will be described.

Quantum computation is based upon physical principles from the theory of quantum mechanics, which is in many ways counterintuitive. Classical computer systems represent a single bit of information deterministically: the value is either logic 0 or logic 1. Quantum computer systems represent a single bit of information as a quantum bit (qubit), which is a unit vector in the complex Hilbert space C^2 [2]. Quantum bit (qubit). In a quantum system the elementary information unit is the qubit. Unlike the classical bit, the qubit does not represent only the value 0 or 1 but a superposition of the two. Its state can be given by [7]:

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (2)$$

where $|0\rangle$ and $|1\rangle$ represent the classical bit values 0 and 1, respectively; α and β are complex numbers such that:

$$|\alpha|^2 + |\beta|^2 = 1 \quad (3)$$

If a superposition is measured with respect to the basis $\{|0\rangle, |1\rangle\}$, the probability to measure $|0\rangle$ is $|\alpha|^2$ and the probability to measure $|1\rangle$ is $|\beta|^2$. Multiple qubits (quantum register)

In classical computing, the possible states of a system of n bits form a vector space of n dimensions, i.e., we have 2^n possible states. However, in a quantum system of n qubits the resulting state space has 2^n dimensions. It is this exponential growth of the state space with the number of particles that suggests a possible exponential speed-up of computation on quantum computers over classical computers. The basis of the state space of a quantum system of n qubits is: $\{|00\dots0\rangle, |00\dots1\rangle, \dots, |11\dots1\rangle\}$ [4, 5].

Measurement: the measurement of a single qubit projects the quantum state onto one of the basis states associated with the measuring device. The result of a measurement is probabilistic and the process of measurement changes the quantum state into the measured one. Multi-qubit measurement can be treated as a series of single-qubit measurements in the standard basis.

Quantum gates: the dynamics of a quantum system are governed by Schrödinger's equation. The transformations must preserve the orthogonality. For a complex vector space, linear transformations that

preserve orthogonality are unitary transformations, defined as follows. Any linear transformation on a complex vector space can be described by a matrix. A matrix M is unitary if $M.M^\dagger = I$. Any unitary transformation of a quantum state space is a legitimate quantum transformation and vice-versa. Rotations are, for example, unitary transformations. One important consequence of the fact that quantum transformations are unitary is that they are reversible. Thus quantum gates, which can be represented by unitary matrices, must be reversible. It has been shown that all classical computations can be done reversibly [7, 10].

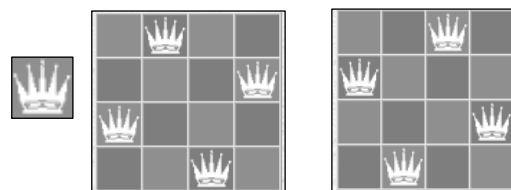
3. The Proposed Approach

Conventional DEAs operate on a set of individuals (chromosomes) forming a population. To be more representative this population must contain a fit number of chromosomes. This makes the solution space very large. So, the classical DEAs are usually very expensive in terms of processing time and memory size. For reducing the number of chromosomes and consequently reducing the heavy computation time, an alternative is proposed: it is the quantum-inspired differential evolution algorithm.

A QDEA is a QDA with quantum coding solutions. This representation will reduce the computation time by decreasing the number of chromosomes. Moreover it gives a better global solution. Like in DEA, initial solutions are encoded in N chromosomes representing the initial population. The difference in a QDEA is that each chromosome does not encode only one solution but all the possible solutions by putting them within a superposition [3, 11].

3.1. Solution Modelling

Every queen on a checker square can reach the other squares that are located on the same horizontal, vertical, and diagonal line. So there can be at most one queen at each horizontal line, at most one queen at each vertical line, and at most one queen at each of the $4n-2$ diagonal lines. Furthermore, since we want to place as many queens as possible, namely exactly n queens, there must be exactly one queen at each horizontal line and at each vertical line [3]. For a 1×1 board, there is one trivial solution: for 2×2 and 3×3 boards, there are no solutions. For a 4×4 board, there are two:



These are considered distinct solutions, even though the solutions are mirror images of each other. There is

no quick and easy way to calculate the total number of NQP solutions for an $N \times N$ board. We can represent the NQP solution by an $N \times N$ matrix A containing only N ones and satisfying the constraint that only one 1 can be in a row, in a column or in a diagonal. For example the matrix below represents a solution of the 4-queens problem:

0	0	1	0
1	0	0	0
0	0	0	1
0	1	0	0

Figure 3. A solution matrix of the 4-queens problem.

3.2. Fitness Function

The penalty of one queen is equal to the number of queens she can check. The fitness of the configuration is equal to the sum of all the queens penalties divided by two, deleting redundancy counting. For example the fitness of the solution presented in Figure 3 is 0: we can easily observe that each queen in this chessboard is the only one in all the lines passing through it; horizontal, diagonal or vertical.

In the matrix solution of Figure 4, the fitness is 8: the penalty of the queen in column 1 is 3: it can check the second, the third and the fourth queen in the same line, the second column's queen has a penalty of 4: it can check the 3 other queens of the same line and that situated in the fourth column of the third line. The third queen can check the queens of the same line, its penalty is 3. Similarly to the second queen in its line, the fourth queen of that first line can check the three other queens in this line and that in the fourth column in the third line. It has a penalty of 4. Finally, the queen in the third line of column 4 can check 2 other queens. So, its penalty is 2. The sum of these penalties is equal to 16, and when dividing it by 2 to reduce redundancy, we will have 8 which is the fitness of this matrix solution.

1	1	1	1
0	0	0	0
0	0	0	1
0	0	0	0

Figure 4. A bad solution matrix of the 4-queens problem.

3.3. Quantum Representation

The solution representation given above can make the search space representation in a conventional DEA very large. Because of this, we propose another representation of the solution, the chromosome, using the quantum state superposition concept. A quantum encoding offers a powerful mean to represent the solution space and reduces by the way the required number of chromosomes [3].

We have represented the solution by a quantum matrix which is equivalent to a chromosome in a

conventional DEA. For example, the following matrix is a quantum matrix representing a 4×4 qubits. Such as, every qubit represents a superposition of 1 and 0 states. So this matrix represents a superposition of all solutions including incorrect solutions.

	0.4134	0.8435	0.8597	0.6633
	0.9105	0.5371	0.5109	0.7484
	0.9759	0.1819	0.6313	0.2524
A Qubit	0.2182	0.9833	0.7755	0.9676
	0.5261	0.8530	0.7707	0.8326
	0.8504	0.5219	0.6372	0.5539
	0.5035	0.2984	0.2579	0.1440
	0.8640	0.9544	0.9662	0.9896

Figure 5. A quantum solution matrix.

The quantum solution matrix is a $(2 \times N) \times N$ matrix which represents the superposition of all possible matrix containing zeros and ones. Each qubit in this solution satisfies the constraints of equation 3: the sum of the squares of the two elements of the qubit is equal to 1.

3.4. The Proposed Algorithm

During the whole process we keep in memory the global best solution. The algorithm consists on applying cyclically the following quantum inspired and differential evolution operations: the first operation is a quantum interference which allows a shift of each qubit in all the direction of the corresponding bit value in the best solution. That is performed by applying a unitary quantum operator which achieves a rotation, whose angle is function of α_i , β_i and the value of the corresponding bit in the best solution [3].

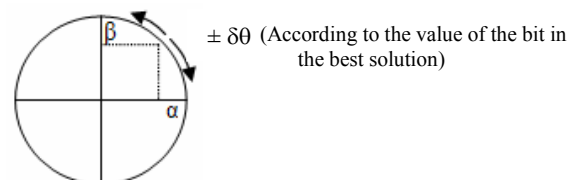


Figure 6. Quantum interference.

The second operation consists of a quantum differential mutation which will perturb values of quantum chromosomes using a difference between two other quantum chromosomes. This difference is multiplied by another factor F before being added to the perturbed chromosome. In this work we have chosen $F=1/1000$.

This operator replaces the quantum-inspired crossover and the quantum-inspired mutation [3], because it has been proved that only the use of differential mutation in a conventional DEA do produce the diversity and the convergence at the same time. So, in this work we will only use the quantum differential mutation.

Table 1. Lookup table of the rotation angle.

θ	ϕ	Reference bit value	Angle
> 0	> 0	1	$+\pi$
> 0	< 0	0	$-\pi$
< 0	> 0	1	$-\pi$
< 0	< 0	0	$+\pi$
> 0	> 0	0	$+\pi$
> 0	< 0	1	$-\pi$
< 0	> 0	1	$-\pi$
< 0	< 0	0	$+\pi$
< 0	< 0	0	$-\pi$

An example is given below; a1, a2, a3 are three chromosomes of the 2-queens problem.

a1=
0.5761 0.1375
0.8173 0.9905
0.7903 0.5635
0.6127 0.8261

a2=
0.3706 0.0400
0.9288 0.9992
0.6795 0.8466
0.7337 0.5322

a3=
0.6887 0.5852
0.7250 0.8109
0.4772 0.0531
0.8788 0.9986

After applying the differential evolution operator: $a3 = a3 + (a1 - a2)/1000$. The chromosome a3 becomes:

a3=
0.6889 0.5853
0.7249 0.8109
0.4773 0.0528
0.8787 0.9989

Finally, we perform a selection of m chromosomes among the n existing in the current generation. For this, we apply first a measurement on each chromosome to have from it one solution among all those present in superposition. But unlike pure quantum systems, the measurement here does not destroy the states' superposition. Since our algorithm operates on conventional computer and does not require the presence of a quantum machine, it is possible and in our interest to keep all the possible solutions in the superposition for the next iterations. For each measurement result, we extract a distribution of N-queens. To evaluate the quality of a solution, we compute its fitness. The best solution is that having the minimal fitness (0 in the optimum).

4. Results and Discussion

This algorithm has been tested on some instances of the N-queens problem: for $N=1$ to 15 with 10 executions for each instance. The obtained results are very promising in comparison with other approaches

such as Classical Genetic Algorithms (CGA) and Hopfield Neural Network (HNN). Compared with the CGA, a QEDA offers a faster convergence with a very limited number of chromosomes. Also, the best local and the global solution evolve in higher performance as shown in Figure 7. Compared with the hopfield neural networks the QDEA give an optimal solution in a limited number of iterations; QDEA are less susceptible to fall in a local optimum. And compared with QGA Figure 8, the search process in the QDEA is more organized and well guided: the best local solutions are generally neighbours of the best global solutions.

Also the complexity of this QDEA is considerably smaller than the CGAs which operate on a large and less representative population, and even than that of the DEA (because of the use of simple, and less number of operators). So, QDEAs are less expensive in representation and processing.

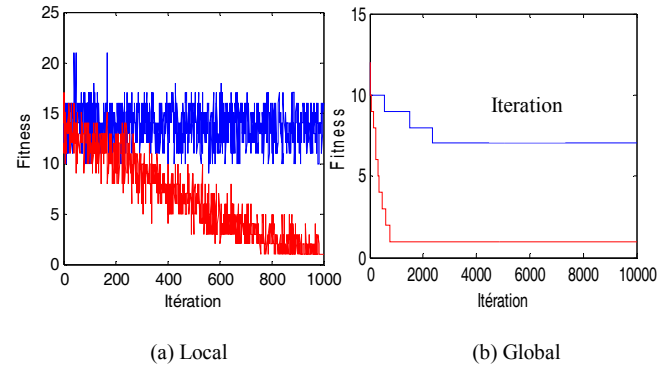


Figure 7. Local and global best solution evolution CGA and in a QDEA.

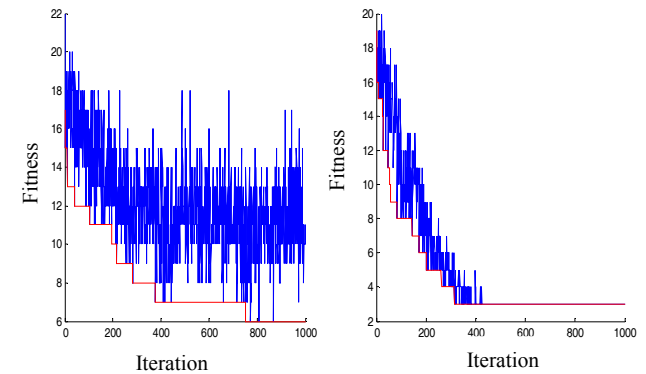


Figure 8. Local best solution evolution and global best solution evolution in a QGA (left) and in a QDEA (right).

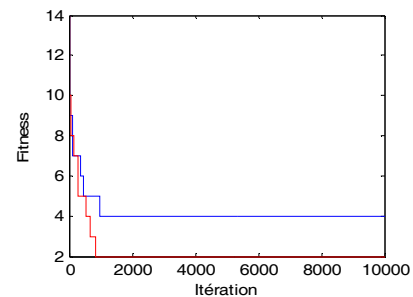


Figure 9. Global best solution evolution in a QGA and in a QDEA.

5. Conclusion

In this paper, we have presented a new approach for solving the NQP. A quantum-inspired differential evolution algorithm has been employed. This algorithm is a hybridization of two well known algorithms. The first is the DEA that uses the differential evolution operator as alternative of classical GAs' mutation. The second is the QGA, proposed for the first time in the works of Han [4] and which was spread in an exponential manner to touch many fields. A QGA has been used for solving the N-Queens problem [3] and has given promising results. Compared with CGA [4, 5] and QGA, our hybrid algorithm has given better results either in term of computation time or in term of fitness evolution. To conclude, we can say that the quantum-inspired differential evolution algorithm is well situated to be among the good alternatives to solve combinatorial optimization problems, especially in term of efficiency and algorithmic complexity. As future work, we will try to apply this approach for solving other combinatorial optimization problems.

References

- [1] Cengiz E., Sarkeshik S., and Murat T., "Different Perspectives of the N-Queens Problem," in *Proceedings of ACM 1992 Computer Science Conference*, Kansas City, pp. 12-15, 1992.
- [2] Draa A., Talbi H., and Batouche M., "A Quantum Inspired Differential Evolution Algorithm for Rigid Image Registration," in *Proceedings of the International Conference on Computational Intelligence*, Istanbul, pp. 408-411, 2004.
- [3] Draa A., Talbi H., and Batouche M., "A Quantum Inspired Genetic Algorithm for Solving the N-Queens Problem," in *Proceedings of the 7th International Symposium on Programming and Systems*, Algiers, pp. 145-152, 2005.
- [4] Han K. and Kim J., "Genetic Quantum Algorithm and its Application to Combinatorial Optimization Problem," in *Proceedings of the 2000 Congress on Evolutionary Computation*, USA, pp. 1354-1360, 2000.
- [5] Han K. and Kim J., "Quantum Inspired Evolutionary Algorithm for a Class of Combinatorial Optimization," *Computer Journal of IEEE Transactions on Evolutionary Computation*, vol. 6, no. 6, pp. 55-59, 2000.
- [6] Karaboga D. and Okdem S., "A Simple and Global Optimization Algorithm for Engineering Problems: Differential Evolution Algorithm," *Computer Journal of Turk Electronic Engineering*, vol. 12, no. 1, pp. 128-132, 2004.
- [7] Rieffel E. and Polak W., "An Introduction to Quantum Computing for Non-Physicists," arxiv.org/quant-ph/9809016v2, 2000.
- [8] Salomon M., "Etude de Parallelisation de Méthodes Heuristiques d'Optimisation Combinatoire Application au Recalage d'Images médicales," *Doctorate Thesis*, Luis Pasteur University, 2001.
- [9] Shor P., "Algorithms for Quantum Computation: Discrete Logarithms and Factoring," in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, Mexico, pp. 187-198, 1994.
- [10] Shor B., "Quantum Computing," *Documenta Mathematica*, <http://east.camel.math.ca/EMIS/journals/DMJDMV/xvolicm/00/Shor.MAN.html>, 1998.
- [11] Talbi H., Draa A., and Batouche M., "A Quantum Inspired Genetic Algorithm for Multi Source Affine Image Registration," in *Proceedings of the International Conference on Image Analysis and Recognition*, Porto, pp. 147-154, 2004.
- [12] Watkins J., *Across the Board: The Mathematics of Chess Problems*, Princeton, 2004.



Amer Draa received his engineer and Master degrees in computer science in 2002 and 2004, respectively, from Mentouri University, Algeria. Currently, he is working toward the PhD degree in computer science. His research interests are in the areas of complex adaptive systems and computer vision.



Souham Meshoul received engineer, MSc and PhD (State Doctorate) degrees in computer science from University of Algeria, in 1990, 1996, and 2004, respectively. Her current research interests include nature inspired computing, bioinformatics, and image processing.



Hichem Talbi received his engineer and Master degrees in computer science in 2000 and 2003, respectively, from Mentouri University, Algeria. Currently, he is about getting the PhD degree in computer science.



Mohamed Batouche received his engineer degree in computer science at University of Constantine, Algeria, MSc and PhD degrees in computer science at University of Nancy, France in 1988, 1989 and 1993, respectively. His research interests are in the field of complex systems, artificial intelligence, and nature inspired computing.