Quantum Inspired Cuckoo Search Algorithm for Graph Coloring

Halima Djeloul, Abdesslem Layeb, Salim Chikhi
MISC Lab, Department of Information & applications.
University of Constantine II,
Ali Mendjeli city, Constantine, 25000, Algeria.
E-mail: halima_djelloul@yahoo.fr
E-mail: layeb@umc.edu.dz
E-mail: chikhi@umc.edu.dz

Abstract

The graph coloring problem (GCP) is one of the most interesting, studied and difficult combinatorial optimization problems. That is why, several approaches were developed for solving this problem, including exact approaches, heuristic approaches, metaheuristics and hybrid approaches. In this paper, we try to solve the graph coloring problem using a new approach based on the quantum inspired cuckoo search algorithm. The first contribution consists in defining an appropriate quantum representation based on qubit representation to represent the graph coloring solutions. The second contribution is the proposition of a novel measure operator based on the adjacency matrix. The third contribution involves the proposition of an adapted hybrid quantum mutation operation. To show the feasibility and the effectiveness of the algorithm, we have used the standard DIMACS benchmark, and the obtained results are very encouraging.

Keywords: Graph Coloring Problem, Cuckoo Search Algorithm, Quantum Computing, Heuristics, Hybrid Algorithms.

1. Introduction

The Graph Coloring Problem (GCP) is one of the most interesting, studied and difficult combinatorial optimization problems. The GCP consists in coloring each vertex of a given graph by using a minimum number of colors called chromatic number [1], so that no two adjacent vertices are colored with the same color. Unfortunately, GCP has been shown to be NP-hard [2], hence, several approaches were developed to handle this problem, that we can classify into three classes; exact approaches, heuristic approaches and metaheuristics approaches. There exist quite a few exact approaches for this problem; they are generally based on the implicit enumeration algorithms [3], as well as branch-and-bound algorithm and its variants [4, 5]. On the other hand, the constructive approaches have been widely proposed to solve the graph coloring problem. The constructive approaches progressively build the
solution, in this category we can cite the algorithm developed by Welsh and Powell [6], the degree of saturation (DSATUR) [7], and the recursive largest first algorithm (RLF) [8]. Moreover, many kinds of metaheuristics and their hybridizations have been used to solve GCP like the tabu Search of Hertz and de Werra[9], Simulated annealing suggested by Johnson et al[10], genetic algorithms [11], ant colony [12], variable neighborhood search VNS [13], Variable Search Space (VSP) that is an extension of the VNS [14], a memetic algorithm by Zhipeng and Jin-Kao Hao [15], a hybrid Artificial Bee Colony Algorithm for Graph 3-Coloring[16], etc.

Far from the graph coloring problem, Cuckoo Search (CS) is an optimization algorithm developed by Xin-She Yang and Suash Deb in 2009 [17]. It was inspired by the obligate brood parasitism of some cuckoo species by laying their eggs in the nests of other host birds (of other species). Some bird’s host can involve direct conflicts with the intruding cuckoos. For example, if a bird’s host discovers that the eggs are strange eggs, it will either throw these alien eggs away or simply abandon its nest and build a new nest elsewhere [18]. The cuckoo’s behavior and the mechanism of Lévy flights [19, 20] have led to the design of an efficient inspired algorithm performing optimization search. The recent applications of Cuckoo Search for optimization problems have shown its promising effectiveness. Moreover, promising discrete cuckoo search algorithms are recently proposed to deal with discrete problems [21, 22, 23].

Quantum Computing (QC) is a new research field that induced intense researches in the last decade, and that covers investigations on quantum computers and quantum algorithms [24]. QC relies on the principles of quantum mechanics like qubit representation and superposition of states. QC is able of processing huge numbers of quantum states simultaneously in parallel. QC brings new philosophy to optimization due to its underlying concepts. Recently, a growing theoretical and practical interest is devoted to researches on merging evolutionary computation and quantum computing [25, 26]. The aim is to get benefit from quantum computing capabilities to enhance both efficiency and speed of classical evolutionary algorithms. This has led to the design of several quantum inspired algorithms such as quantum inspired genetic algorithm [25], quantum differential algorithm [27], quantum inspired scatter search [28], etc. Unlike pure quantum computing, quantum inspired algorithms don’t require the presence of a quantum machine to work. Quantum inspired algorithms have been used to solve successfully many combinatorial optimization problems [25, 29]. Recently a new hybrid algorithm called Quantum Inspired Cuckoo Search algorithm
(QICSA) is proposed to cope with combinatorial optimization problems [21]. The proposed algorithm combines Cuckoo Search algorithm and quantum computing in new one. The features of the proposed algorithm consist in adopting a quantum representation of the search space. The other feature of QICSA is the integration of the quantum operators in the cuckoo search dynamics to enhance the optimization capacities of the basic cuckoo search algorithm.

The present study was designed to investigate the use of the QICSA algorithm to deal with the graph coloring problem. The main features of the proposed approach consist in adopting a quantum representation of the search space and the integration of the quantum operators like interference, measure and mutation in the cuckoo search dynamics in order to minimize the number of colors. Moreover, we have modified the standard quantum measure operation by introducing the adjacency matrix during the operation of measurement. In addition, we have combined the two mutation methods proposed in [30] to replace the standard quantum mutation operation. Finally, we have tested our algorithm on some DIMACS instances taken from (http://mat.gsia.cmu.edu/COLOR/instances.html) and the results found are promising.

The reminder of the paper is organized as follows. In section 2, a formulation of the tackled problem is given. In section 3, an overview of quantum computing is presented. In section 4, the cuckoo search algorithm presented. The proposed method is described in section 5. Experimental results are discussed in section 6. Finally, conclusions and future work are drawn.

2. Problem formulation

Graph Coloring Problem (GCP) is a well-known combinatorial problem, and important task in solving many real problems such as the frequency assignment problem [31], crew scheduling [32], register allocation [33], etc. A graph is k-colorable if and only if it can be colored using k colors. Formally, a k-coloring will be represented by a set \( S = \{C(v_1), C(v_2), \ldots, C(v_n)\} \) such as \( C(v_i) \) is the color assigned to the vertex \( v_i \). If for all \( \{u,v\} \in E, C(u) \neq C(v) \), then \( S \) is a legal k-coloring; otherwise, \( S \) is an unfeasible k-coloring. In the optimization version of the graph coloring problem, the principal objective is to minimize the total number of colors used to color a given graph.

Formally, the graph coloring problem can be formulated as follows:

Given a k-coloring \( S = \{C(v_1), C(v_2), \ldots, C(v_n)\} \) with the set \( V = \{v_1, \ldots, v_n\} \) of vertices, the evaluation function \( f \) counts the number of conflicting vertices produced by \( S \) such that:
\[ f(S) = \sum_{uv \in E} \delta_{uv} \]  

Where:

\[ \delta_{uv} = \begin{cases} 
1, & \text{if } C(u) = C(v) \\
0, & \text{otherwise.} 
\end{cases} \]

By consequent, a coloring \( S \) with \( f(S) = 0 \) corresponds to a feasible \( k \)-coloring.

### 3. Overview of Quantum Computing

Quantum computing is a new theory which has emerged as a result of merging computer science and quantum mechanics. Its main goal is to investigate all the possibilities a computer could have if it followed the laws of quantum mechanics. The origin of quantum computing goes back to the early 80’s when Richard Feynman observed that some quantum mechanical during the last decade, quantum computing has attracted widespread interest and has induced intensive investigations and researches since it appears more powerful than its classical counterpart. Indeed, the parallelism that the quantum computing provides reduces obviously the algorithmic complexity. Such an ability of parallel processing can be used to solve combinatorial optimization problems which require the exploration of large solutions spaces. The basic definitions and laws of quantum information theory are beyond the scope of this paper. For in-depth theoretical insights, one can refer to [24].

The qubit is the smallest unit of information stored in a two-state quantum computer. Contrary to classical bit which has two possible values, either 0 or 1, a qubit will be in the superposition of those two values. The state of a qubit can be represented by using the bracket notation:

\[ |\psi\rangle = a|0\rangle + b|1\rangle \]

where \( |\psi\rangle \) denotes more than a vector \( \vec{\psi} \) in some vector space, \( |0\rangle \) and \( |1\rangle \) represent the classical bit values 0 and 1 respectively; \( a \) and \( b \) are complex numbers such that:

\[ |a|^2 + |b|^2 = 1 \]

\( a \) and \( b \) are complex number that specify the probability amplitudes of the corresponding states. When we measure the qubit’s state we may have ‘0’ with a probability \( |a|^2 \) and we may have ‘1’ with a probability \( |b|^2 \).
A system of m-qubits can represent $2^n$ states at the same time. Quantum computers can perform computations on all these values at the same time. It is this exponential growth of the state space with the number of particles that suggests exponential speed-up of computation on quantum computers over classical computers. Each quantum operation will deal with all the states present within the superposition in parallel. When observing a quantum state, it collapses to a single state among those states.

Quantum Algorithms consist in applying successively a series of quantum operations on a quantum system. Quantum operations are performed using quantum gates and quantum circuits. It should be noted that designing quantum algorithms is not easy at all. Yet, there is not a powerful quantum machine able to execute the developed quantum algorithms. Therefore, some researchers have tried to adapt some properties of quantum computing in the classical algorithms. Since the late 1990s, merging quantum computation and evolutionary computation has been proven to be a productive issue when probing complex problems. Like any other EA, a Quantum Evolutionary Algorithm (QEA) relies on the representation of the individual, the evaluation function and the population dynamics. The particularity of QEA stems from the quantum representation they adopt which allows representing the superposition of all potential solutions for a given problem. It also stems from the quantum operators it uses to evolve the entire population through generations. QEA has been successfully applied on many problems [25, 26, 21].

4. Cuckoo Search Algorithm

In order to solve complex problems, ideas gleaned from natural mechanisms have been exploited to develop heuristics. Nature inspired optimization algorithms have been extensively investigated during the last decade paving the way for new computing paradigms. The ultimate goal is to develop systems that have ability to learn incrementally, to be adaptable to their environment and to be tolerant to noise. Several nature inspired optimization algorithms were developed in the last decade such as Neural Networks [34], Particle Swarm Optimization [35, 36], Ant Colony Optimization [37,38], Artificial Plant Optimization Algorithm [39], Gravitational Search Algorithm [40], etc. One of the recent developed bioinspired algorithms is the Cuckoo Search (CS) [17] which is based on lifestyle of Cuckoo bird. Cuckoos use an aggressive strategy of reproduction that involves the female hack nests of other birds to lay their eggs fertilized. Sometimes, the egg of cuckoo in the nest is discovered and the hacked birds discard or abandon the nest and start their own brood
elsewhere. The Cuckoo Search proposed by Yang and Deb 2009 [17] is based on the following three idealized rules:

- Each cuckoo lays one egg at a time, and dumps it in a randomly chosen nest;
- The best nests with high quality of eggs (solutions) will carry over to the next generations;
- The number of available host nests is fixed, and a host can discover an alien egg with a probability $p_a \in [0, 1]$. In this case, the host bird can either throw the egg away or abandon the nest so as to build a completely new nest in a new location.

The last assumption can be approximated by a fraction $p_a$ of the $n$ nests being replaced by new nests (with new random solutions at new locations). The generation of new solutions $x(t+1)$ is done by using a Lévy flight (eq.5). Lévy flights essentially provide a random walk while their random steps are drawn from a Lévy distribution for large steps which has an infinite variance with an infinite mean (eq.6). Here the consecutive jumps/steps of a cuckoo essentially form a random walk process which obeys a power-law step length distribution with a heavy tail [17].

$$x_i^{t+1} = x_i^t + \alpha \oplus \text{Lévy}(\lambda)$$  \hspace{1cm} (5)

$$\text{Lévy} \sim u=t^{-\lambda}$$  \hspace{1cm} (6)

Where $\alpha > 0$ is the step size which should be related to the scales of the problem of interest. Generally we take $\alpha = O(1)$. The product $\oplus$ means entry-wise multiplications. This entry-wise product is similar to those used in PSO, but here the random walk via Lévy flight is more efficient in exploring the search space as its step length is much longer in the long run.

The main characteristic of CS algorithm is its simplicity. In fact, comparing with other population or agent-based metaheuristic algorithms such as particle swarm optimization and harmony search, there are few parameters to set. The applications of CS into engineering optimization problems have shown its encouraging efficiency. For example, a promising discrete cuckoo search algorithm is recently proposed to solve nurse scheduling problem [41]. In [22], a binary version of cuckoo search is proposed to solve the knapsack problems. An improved hybrid Cuckoo Search algorithm for permutation flow shop scheduling problems is presented in [23]. An efficient computation approach based on cuckoo search has been proposed for data fusion in wireless sensor networks [42]. A recent cuckoo search algorithm
was proposed for coloring the planar graph [43]. Finally and not last, a new model based on cuckoo search algorithm is proposed for estimating the test efforts [44].

In more details, the proposed cuckoo search algorithm can be described as follow:

| **Objective function** \( f(x), x = (x_1, \ldots, x_d)^T \) |
| **Initial a population** of \( n \) host nests \( x_i (i = 1, 2, \ldots, n) \) |
| **while** (\( t < \text{MaxGeneration} \)) or (stop criterion); |
| \( \bullet \) Get a cuckoo (say i) randomly by Lévy flights; |
| \( \bullet \) Evaluate its quality/fitness \( F_i \); |
| \( \.outline{\bullet} \) Choose a nest among \( n \) (say j) randomly; |
| \( \bullet \) if \( (F_i > F_j) \), |
| Replace j by the new solution; |
| end |
| \( \bullet \) Abandon a fraction (\( p_a \)) of worse nests |
| \( \bullet \) build new ones at new locations via Lévy flights; |
| \( \bullet \) Keep the best solutions (or nests with quality solutions); |
| \( \bullet \) Rank the solutions and find the current best; |
| **end while** |

**Figure 1.** Cuckoo Search Schema.

### 4.1. Quantum Inspired Cuckoo Search

In this section, we present the Quantum Inspired Cuckoo Search (QICSA) which integrates the quantum computing principles such as qubit representation, measure operation and quantum mutation, in the core of the cuckoo search algorithm. This proposed model will focus on enhancing diversity and the performance of the cuckoo search algorithm [21].

The QICSA architecture, which has been developed to solve combinatorial problems, is explained in Figure 2. Our architecture contains three essential modules. The first module contains a quantum representation of cuckoo swarm. The particularity of quantum inspired cuckoo search algorithm stems from the quantum representation it adopts and which allows representing the superposition of all potential solutions for a given problem. Moreover, the generation of a new cuckoo depends on the probability amplitudes \( a \) and \( b \) of the qubit function \( \Psi \) (eq.3). The second module contains the objective function and the selection operator. The selection operator is similar to the elitism strategy used in genetic algorithms. Finally, the third module, which is the most important, contains the main quantum cuckoo dynamics. This module is composed of 4 main operations inspired from quantum computing and cuckoo search algorithm: Measurement, Mutation, Interference, and Lévy flights operations. QICSA uses these operations to evolve the entire swarm through generations [21].
5. The proposed approach for solving the graph coloring problem

The development of the suggested approach called QICSA.Col is based mainly on a quantum representation of the search space associated with the problem, and a QICSA dynamic used to explore this space by using inspired quantum operations. In order to show how quantum computing concepts have been tailored to the problem at hand, we need first to derive a representation scheme which includes the definition of an appropriate quantum representation of potential graph coloring solutions and the definition of quantum operators. Then, we describe how these defined concepts have been integrated in cuckoo Search algorithm.

5.1. Quantum representation of graph coloring solution

In order to apply easily quantum principles on graph coloring problem, we need to map the graph coloring solutions into a quantum representation that could be easily manipulated by quantum operators. So, the graph coloring solution is represented as binary matrix (Figure.5) Satisfying the following criteria:

- For a graph of N nodes and K colors, the size of the binary matrix is K*N.
  The columns represent the nodes and the rows represent the colors.
- The presence of 1 in the position \((i,j)\) indicates that the node \(j\) colored with the color \(i\).
In each column there is a single 1, i.e. the node is colored with one color.

Figure 3 mentions a graph of 9 nodes, a feasible K-coloring for this graph is given in Figure 4, and the corresponding binary matrix is given in Figure 5. According to the example the nodes 1, 4, 8 are assigned the first color, the nodes 2, 3, 6 and 7 are colored with the second color and the nodes 5, 9 are assigned the third color.

Figure 3. Uncolored graph

Figure 4. Feasible coloring for the graph

\[
\begin{pmatrix}
1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1
\end{pmatrix}
\]

Figure 5. Binary representation of the solution

In terms of quantum computing, each class of colors is represented as a quantum register as shown in Figure 6. One quantum register contains a superposition of all possible vertices colored by a given color. Each column \( \begin{pmatrix} a \\ b \end{pmatrix} \) represents a single qubit and corresponds to the binary digit 1 or 0. The probability amplitudes \( a_i \) and \( b_i \) are real values satisfying \( |a|^2 + |b|^2 = 1 \). For each qubit, a binary value is computed according to its probabilities \( |a|^2, |b|^2 \) which can be interpreted as the probabilities to have respectively 0 or 1. Consequently, all possible graph coloring solutions can be represented by one quantum matrix \( QM \) (Figure 7) that contains the superposition of all possible solutions. This quantum matrix can be viewed as a probabilistic representation of all potential graph coloring solutions. When embedded within a cuckoo search algorithm framework, it plays the role of a nest. By consequent, a quantum representation offers a powerful way to represent the solution space and reduces consequently
the required number of cuckoo. Only one quantum matrix is needed to represent the entire swarm.

\[
\begin{pmatrix}
  a_1 & a_2 & \cdots & a_m \\
  b_1 & b_2 & \cdots & b_m
\end{pmatrix}
\]

**Figure 6.** Quantum register encoding a row in the binary matrix.

\[
\begin{pmatrix}
  a_{11} & a_{12} & \cdots & a_{1m} \\
  b_{11} & b_{12} & \cdots & b_{1m} \\
  a_{21} & a_{22} & \cdots & a_{2m} \\
  b_{21} & b_{22} & \cdots & b_{2m} \\
  \vdots & \vdots & \ddots & \vdots \\
  a_{n1} & a_{n2} & \cdots & a_{nm} \\
  b_{n1} & b_{n2} & \cdots & b_{nm}
\end{pmatrix}
\]

**Figure 7.** Quantum representation of graph coloring solution

### 5.2. Constructive heuristic for generating the initial population

For generating the initial population, we have used a modified version of the constructive RLF algorithm proposed in [45], which colors the vertices one node at a time, in the following greedy way:

Let \( N_{cl} \) to be the next color to be assigned, \( Y \) the set of uncolored vertices that can be assigned to the color \( N_{cl} \), and \( B \) the set of uncolored vertices that cannot be assigned to the color \( N_{cl} \).

- If \( N_{cl} > k \) then the remaining nodes will be colored randomly using the set of colors \( \{1, 2, \ldots, k\} \).
- Otherwise, choose the first vertex \( v \in Y \) randomly, color \( v \) with the color \( N_{cl} \), and move its neighbors included in \( Y \) from \( Y \) to \( B \).

In more details, the heuristic used can be described as follow:
5.3. Quantum operators
The QICSA algorithm contains some quantum operations like measurement, interference and mutation operators. This integration helps to increase the optimization capacities of the cuckoo search, because the levy flight operator is not sufficient to optimize hard problems.

5.3.1. Measurement
This operation transforms by projection the quantum vector into a binary vector (Figure 9). Therefore, there will be a solution among all the solutions present in the superposition. But contrary to the pure quantum theory, this measurement does not destroy the superposition. that has the advantage of preserving the superposition for the following iterations knowing that we operate on traditional machines. The binary values for a qubit are computed according to their probabilities $|a|^2$ and $|b|^2$.

For the graph coloring problem, this operation is accomplished as follows: for each qubit, we generate a random number $Pr$ between 0 and 1; the value of the corresponding bit is 1 if the value $|b|^2$ is greater than $Pr$, and otherwise the bit value is 0. However, the use of the standard measure operation defined in [21] can lead to infeasible solutions, and then increase the computational time of the algorithm to find good solutions. Indeed, by using the standard

<table>
<thead>
<tr>
<th>B = φ , Y = V, Ncl = 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>While (Y ≠ φ) do</td>
</tr>
<tr>
<td>Ncl = Ncl + 1</td>
</tr>
<tr>
<td>If (Ncl &gt; k)</td>
</tr>
<tr>
<td>Color the remaining vertices with colors randomly chosen in the set {1, 2, . . . , k}</td>
</tr>
<tr>
<td>else choose randomly v ∈ Y</td>
</tr>
<tr>
<td>C(v) = Ncl , B = B ∪ N_{G,Y}(v) , Y = Y \ ({v} ∪ N_G(v))</td>
</tr>
<tr>
<td>While (Y ≠ φ)</td>
</tr>
<tr>
<td>Choose v ∈ Y</td>
</tr>
<tr>
<td>C(v) = Ncl , B = B ∪ N_{G,Y}(v) , Y = Y \ ({v} ∪ N_G(v))</td>
</tr>
<tr>
<td>Endwhile</td>
</tr>
<tr>
<td>Endif</td>
</tr>
<tr>
<td>Endwhile</td>
</tr>
<tr>
<td>End_InitPop</td>
</tr>
</tbody>
</table>

Figure 8. Constructive modified RLF [45]
measure, we can get an uncolored node or a node having more than one color. In order to delete this kind of solutions, we have introduced the adjacency matrix in the measure operation. So, the value 1 is obtained if the value $|b|^2$ is greater than the random number $P_r$, and any neighbor of the current node have already the same color. To correct the uncolored vertex, we have introduced a randomly assignment of colors to the remaining vertices in the core of the measure operation.

It should be noted that, the measurement operation can be seen also as a diversification operator. Indeed, two successive measurements do not give necessarily the same solution which increases the diversification capacities of our approach. The obtained binary vector is then translated into the problem solution. Consequently, with one quantum matrix, we can get several different solutions for the graph coloring problem, which gives a great diversity to the cuckoo search algorithm.

$$\begin{pmatrix} a_1 & a_2 & \cdots & a_n \\ b_1 & b_2 & \cdots & b_n \end{pmatrix} \xrightarrow{\text{Measure}} (0,1,\ldots,1)$$

**Figure 9.** Quantum measurement.

### 5.3.2. Quantum interference

This operation amplifies the amplitude of the best solution and decreases the amplitudes of the bad ones. It primarily consists in moving the state of each qubit in the direction of the corresponding bit value in the best solution in progress. The operation of interference is useful to intensify research around the best solution and it plays the role of local search method. This operation can be accomplished by using a unit transformation which achieves a rotation whose angle is a function of the amplitudes $a_i, b_i$ (figure 10). The rotation angle’s value $\delta \theta$ should be well set in order to avoid premature convergence. A big value of the rotation angle can lead to premature convergence or divergence; however a small value to this parameter can increase the convergence time. Consequently, the angle is set experimentally and its direction is determined as a function of the values of $a_i, b_i$ and the corresponding element’s value in the binary vector (table1). In our algorithm, we have set the rotation angle $\delta \theta = \pi/20$. However, we can use a dynamic value of the rotation angle in order to increase the performance of the interference operation.
This operator is inspired from the evolutionary mutation. It allows moving from the current solution to one of its neighbors. This operator allows exploring new solutions and thus enhances the diversification capabilities of the search process. There are several types of quantum inspired mutations that can be integrated in the core of the cuckoo search algorithm:

- **Inter-qubit Mutation**: this operator performs permutation between two qubits. It consists first in selecting randomly a register in the quantum matrix. Then, pairs of qubits are chosen randomly according to a defined probability (Figure 11).
- **Intra-qubit Mutation**: it consists in selecting randomly a qubit according to a defined probability, next we make a permutation between the qubit amplitudes $a_i$ et $b_i$ as it shown on figure 12.

### Table 1. Lookup table of the rotation angle

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>Reference bit value</th>
<th>angle</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;0</td>
<td>&gt;0</td>
<td>1</td>
<td>$+\delta\theta$</td>
</tr>
<tr>
<td>&gt;0</td>
<td>&gt;0</td>
<td>0</td>
<td>$-\delta\theta$</td>
</tr>
<tr>
<td>&gt;0</td>
<td>&lt;0</td>
<td>1</td>
<td>$-\delta\theta$</td>
</tr>
<tr>
<td>&gt;0</td>
<td>&lt;0</td>
<td>0</td>
<td>$+\delta\theta$</td>
</tr>
<tr>
<td>&lt;0</td>
<td>&gt;0</td>
<td>1</td>
<td>$-\delta\theta$</td>
</tr>
<tr>
<td>&lt;0</td>
<td>&gt;0</td>
<td>0</td>
<td>$+\delta\theta$</td>
</tr>
<tr>
<td>&lt;0</td>
<td>&lt;0</td>
<td>1</td>
<td>$+\delta\theta$</td>
</tr>
<tr>
<td>&lt;0</td>
<td>&lt;0</td>
<td>0</td>
<td>$-\delta\theta$</td>
</tr>
</tbody>
</table>

Figure 10. Quantum interference

Figure 11. Inter-qubit quantum mutation.

Figure 12. Intra-qubit quantum mutation.

However, the previous mutations do not give always good solutions, so we have replaced them by one specific mutation obtained by the combination of the two mutation methods (Figure 14, Figure 15) presented in [30]. The following algorithm describes the mutation operator.
The two previous mutation methods are used separately in [30] depending on the best fitness. In our proposed approach, we combine them in one mutation method as it is described in figure 13. The hybrid mutation is applied to the binary solution to get better solutions. The
figure 16 illustrates the process followed by the mutation operator applied to an illegal coloring for the graph shown in figure 3. This latter starts first with checking out the conflicting nodes (in figure 16.a the node 3 is the first conflicting node). Then it changes node’s color by choosing randomly another one from the validColors set (figure 16.b, 16.d). The process will be repeated until no conflicting node cannot be found. In the case of the emptiness of the validColors set, a color chosen randomly from the allColors set = \{1, 2, … , k\} will be assigned to the node (figure 16.c).

![Illustrative schema](image)

**Figure 16.** Illustrative schema

### 5.4. Outlines of the proposed algorithm

Now, we describe how the representation scheme including quantum representation and quantum operators has been embedded within a cuckoo search algorithm and resulted in a hybrid stochastic algorithm used to solve the graph coloring problem.

In the initialization step, the developed algorithm called QICSACOL (Quantum Inspired Cuckoo Search Algorithm for Graph COLoring), generates the initial population of nests by using the RLF algorithm [45], to get diverse solutions, some randomness is inserted to the RLF heuristic.

QICSACOL progresses through a number of generations according to the QICSA dynamics. At each iteration, the following main tasks are performed. A new cuckoo is built using the Lévy flights operator followed by the mutation. The next step is to evaluate the current cuckoo. For that, we apply the measure operation in order to get a binary solution which represents a potential solution. The binary solution is evaluated by using the number of edge violated as objective function. After this step, we apply the interference operation according to the best current solution. We replace some worst nests by the current cuckoo if it is better
or by new random nests generated by the Lévy flight. The selection phase in QICSA of the best nests or solutions is comparable to some form of elitism selection used in genetic algorithms, which ensures the best solution is always kept in the next iteration. If a feasible $k$-coloring is found the number of colors $k$ will be reduced by one, and the procedure will be iterated until a legal $k$-coloring cannot be found after a stopping criterion is met. In more details, the proposed QICSA can be described as in figure 17.

The particularity of the proposed QICSA algorithm for graph coloring problem stems from the quantum characteristics like the superposition, the new measurement operator, and the interference which allow good balance between exploration and the intensification of the search space. Indeed, the use of specific operators like the measurement based on the adjacency matrix, and the hybrid mutation have increased considerably the algorithm’s performance.

<table>
<thead>
<tr>
<th>Input: problem data.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output: problem solution.</td>
</tr>
<tr>
<td>Construct an initial solution of $p$ hosts nest.</td>
</tr>
<tr>
<td><strong>While</strong> (stop criterion)</td>
</tr>
<tr>
<td>- Apply levy flight operator to get cuckoo randomly.</td>
</tr>
<tr>
<td>- <strong>Apply a mutation operator.</strong></td>
</tr>
<tr>
<td>- <strong>Apply measurement operator.</strong></td>
</tr>
<tr>
<td>- Evaluate the quality/fitness of the cuckoo.</td>
</tr>
<tr>
<td>- <strong>Apply interference operator.</strong></td>
</tr>
<tr>
<td>- replace some nests among $n$ randomly by the new solution according to its fitness;</td>
</tr>
<tr>
<td>- A fraction ($p_a$) of the worse nests are abandoned and new ones are built via Lévy flights;</td>
</tr>
<tr>
<td>- Keep the best solutions (or nests with quality solutions);</td>
</tr>
<tr>
<td>- Rank the solutions and find the current best;</td>
</tr>
<tr>
<td><strong>End while</strong></td>
</tr>
</tbody>
</table>

**Figure 17.** Quantum Inspired Cuckoo Search Schema.

### 6. Implementation and validation

QICSA for graph coloring (QICSACOL) is implemented in Matlab 7.9 with Intel core i3 processor and 4 GB of memory. To assess the efficiency of our approach, a set of standard
DIMACS benchmark have been used. The results are given in Table 2, the first column is the name of the instance, the second column contains the number of vertices, the third contains the number of edges, the fourth column contains the chromatic number, the fifth column contains the results of the first version of our approach without the mutation operator (QICSACOL1), the sixth column contains the result of the second version that uses the mutation operator (QICSACOL2), the seventh column contains the result of the final version (QICSACOL3) which uses the modified measurement operation explained in the previous section, the eighth column contains the results of the heuristic used for generating the initial population, and the final columns contain the results of two genetic algorithms proposed in [30] and [46]. Finally, Friedman test is used to compare statistically the results found.

<table>
<thead>
<tr>
<th>Instance</th>
<th>[V]</th>
<th>[E]</th>
<th>(\chi)</th>
<th>QICSACOL1</th>
<th>QICSACOL2</th>
<th>QICSACOL3</th>
<th>InitPop [45]</th>
<th>GA [30]</th>
<th>PGA [46]</th>
</tr>
</thead>
<tbody>
<tr>
<td>mycel3</td>
<td>11</td>
<td>20</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>mycel4</td>
<td>23</td>
<td>71</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>queen5_5</td>
<td>25</td>
<td>160</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>5</td>
<td>7</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>queen6_6</td>
<td>36</td>
<td>290</td>
<td>7</td>
<td>9</td>
<td>8</td>
<td>8</td>
<td>9</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>mycel5</td>
<td>47</td>
<td>236</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>huck</td>
<td>74</td>
<td>301</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>jean</td>
<td>80</td>
<td>254</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>david</td>
<td>87</td>
<td>406</td>
<td>11</td>
<td>12</td>
<td>11</td>
<td>11</td>
<td>12</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>games120</td>
<td>120</td>
<td>638</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>miles250</td>
<td>128</td>
<td>387</td>
<td>8</td>
<td>9</td>
<td>8</td>
<td>8</td>
<td>11</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>miles500</td>
<td>128</td>
<td>1170</td>
<td>20</td>
<td>21</td>
<td>20</td>
<td>20</td>
<td>22</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>anna</td>
<td>138</td>
<td>493</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>fpsol2.i.1</td>
<td>496</td>
<td>11654</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>66</td>
<td>65</td>
<td>65</td>
</tr>
</tbody>
</table>

Table 2. Results on DIMACS graphs.

First, there are clear differences between the different versions of the proposed approach and the best solutions given by the randomized constructive heuristic InitPop (Figure 18). Moreover, QICSACOL3 which uses a modified measurement operator and a specific mutation is the closest to the exact results, followed by the QICSACOL2 and QICSACOL1. According to table 2, the QICSACOL3 was able to reach the optimum in 12 cases out of 13. For the graph queen6_6 the difference was only one. Experiments show the effectiveness of the introduction of mutation in QICSACOL2 and QICSACOL3, since it has improved the results achieved by QICSACOL1 which has not a statistically significant difference compared to the results obtained by InitPop (Figure 18).

The statistical Friedman test of Figure 19 represents a comparison of the Exact, QICSACOL, initPop, GA and PGA results, compared to the genetic algorithm GA [30] and to the parallel
version PGA [46], QICSACOL has the same performances to the PGA and comparable results to GA algorithm which has managed to obtain the optimal results through using a more sophisticated specific operators.

The examination of the results shows that very good results can be obtained with a good construction algorithm, specific mutations, and more sophisticated measure operation in order to avoid exploring unpromising areas in the search space, and so accelerate the algorithm convergence.

Figure 18. Friedman test compares the different version of the proposed algorithm
Figure 19. Friedman test compares the proposed algorithm against genetic algorithms

7. Conclusion

In this work, we have presented a new metaheuristic for the graph coloring problem. The proposed metaheuristic is based on quantum inspired cuckoo search algorithm called QICSACOL. The effectiveness of our approach is explained by the good combination between the performances of the quantum computing and the cuckoo search algorithm, which leads the proposed algorithm to effectively explore the search space and locate good solutions. The main contributions of our approach are the definition of a quantum representation for the graph coloring solution, the use of a new measure operation based on the adjacency matrix, and the use of an adapted mutation operator. The optimization process consists of the application of a cuckoo search dynamics enhanced by quantum operations such as the interference, the quantum mutation and measurement. The approach has been thoroughly assessed with different instance types and problem sizes taken from the DIMACS benchmark. The proposed algorithm reduces efficiently the population size, and the number of iterations to have the optimal solution. Thanks to superposition, interference, the new measure operator and the adapted mutation, better balance between intensification and diversification of the search is achieved.

References


