

A Hybrid Quantum Genetic Algorithm and Local Search based DPLL for Max 3-SAT Problems

Abdesslem Layeb* and Djamel-Eddine Saidouni

MISC Laboratory, Computer Science Department, University Constantine 2, Constantine, Algeria

Received: 26 May. 2013, Revised: 9 Sep. 2013, Accepted: 10 Sep. 2013

Published online: 1 Jan. 2014

Abstract: In this paper, we present a new framework for combining complete and incomplete methods in order to deal with the Max Sat problem. The objective is to find the best assignment for a set of Boolean variables, which gives the maximum of verified clauses in a Boolean formula. Unfortunately, this problem has been shown to be NP-hard (non-deterministic polynomial-time hard) if the number of variables per clause is greater than 3. The proposed approach is based on quantum inspired genetic principles and the well known exact algorithm DPLL. The underlying idea is to harness the optimization capabilities of quantum genetic algorithm to achieve good quality solutions for Max SAT problem. A new local search based on DPLL algorithm has been embodied in the quantum genetic algorithm leading to an efficient hybrid framework which achieves better balance between diversification and intensification search. The obtained results are very encouraging and show the feasibility and effectiveness of the proposed hybrid approach.

Keywords: Quantum Genetic Algorithm, MAX 3-SAT problem, DPLL procedure

1 Introduction

There are several combinatorial optimization problems that are difficult to solve. Among these problems, there is the problem of propositional satisfiability (SAT). It is the task of determining the satisfiability of a given Boolean formula by looking for the variable assignment that makes this formula evaluating to true. In 1971, Stephen Cook had demonstrated that the Max-SAT problem is NP-complete [1]. This complex problem has several applications in different areas such as model checking [2], graph colouring [3] to cite just few. The Maximum Satisfiability (Max-SAT) problem is a variant of SAT problem that aims to find the variable assignment maximizing the number of satisfied clauses Max 3-SAT is special case of Max SAT problems where the Boolean expressions are written in CNF (Conjunctive normal form) with 3 variables per clause. The following Boolean formula is in 3-CNF expression form: $(x1 \vee x2 \vee x3) \wedge (\neg x1 \vee x2 \vee x5) \wedge \dots \wedge (x1 \vee x4 \vee x3)$. There are other variants of the Max-SAT problem such as Weighted Max-SAT [4] and Partial Max-SAT [5]. To solve this problem, many algorithms were proposed. In fact, there are two classes of algorithms for solving instances of SAT in practice: Complete and Incomplete

methods. The complete algorithms are able to verify the satisfiability or the unsatisfiability of the SAT problem, although they have an exponential complexity. They usually have an exponential complexity [6]. The most popular algorithms of this class are based on the Davis-Putnam-Loveland algorithm (DPLL) [7]. The DPLL procedure is based on a systematic search process used for finding a satisfying assignment for a given Boolean formula or verifying that it is unsatisfiable. On the other hand, incomplete methods are principally based on local search and evolutionary algorithms. Incomplete methods find good quality solutions in reasonable time. Therefore, they do not guarantee optimality. This class of methods encompasses Evolutionary Algorithms (EA) [8, 9], Stochastic Local Search (SLS) methods [10, 11] and hybrid methods [12, 13]. One of the iterative methods that have been used recently to solve this type of problem is Genetic Algorithms (GA) [14, 15]. It is a stochastic iterative algorithm that operates on a population of individuals. GA adapts nature optimizing principles like mechanics of natural selection and natural genetics. Each individual represents a potential solution in the search space of the problem. Basically, a genetic algorithm consists of three essential operations: selection, crossover, and mutation. The selection operator consists

* Corresponding author e-mail: layeb.univ@gmail.com

in selecting an intermediate population from the current one in order to create the future population by using crossover and mutation operators. The crossover operator merges two individuals to provide new ones. The mutation operator allows moving each solution to one of its neighbours in order to maintain a good diversity during the optimization process. Although GAs have been used to solve many difficult engineering problems, they have been shown to be inappropriate for solving SAT problems [16]. To overcome this drawback and in order to get better performance and quality convergence, their hybridization with local search method is exploited. Far from SAT problem, Quantum computing is a new research field that covers investigations on quantum mechanical computers and quantum algorithms [17, 18]. QC relies on the principles of quantum mechanics like qubit representation and superposition of states. QC is capable 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 [19–22]. 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 Quantum inspired Genetic Algorithms QGA that have been proven to be better than conventional GAs. Unlike pure quantum computing, QGA does not require the presence of a quantum machine to work. With this background, we propose a new hybrid approach which we call QGADPLL. It is based on a hybrid algorithm of Quantum Inspired Genetic and local search. For that, a problem formulation in terms of quantum representation and evolutionary dynamic borrowing quantum operators were defined. The quantum representation of the solutions allows the coding of all the potential solutions of MAX SAT with a certain probability. The optimization process consists in the application of a quantum dynamic constituted of a set of quantum operations such as interference, quantum mutation and measure improved by the use of local search method in order to well explore the search space. The local search is based on the exact algorithm DPLL. The idea consists in applying of DPLL algorithm on small set of clauses selected randomly from the MAX SAT problem formula. In order to increase the performance of the local search, the subset of clauses must contain at least one false clause. The experiences carried out on QGADPLL showed the feasibility and the effectiveness of our approach. Consequently, the remainder of the paper is organized as follows: Section 2 presents the problem formulation. Section 3 presents a state of art on the Max sat solvers. A brief introduction to quantum computing is presented in Section 4. The proposed approach is described in Section 5. Section 6 illustrates some experimental results. Then, we terminate by giving conclusion and some perspectives.

2 Problem formulation

Throughout this paper, n will represent the number of Boolean variables and m will denote the number of clauses in formula F . Given a Boolean formula F expressed in CNF (Conjunctive Normal Form), let have n Boolean variables x_1, x_2, \dots, x_n , and m clauses C_1, C_1, \dots, C_m . The problem of MAX SAT can be formulated as follow:

- An assignment to those variables is a vector: $v = (v_1, v_2, \dots, v_n) \in \{0, 1\}^n$
- A clause C_i of length k is a disjunction of k literals, $C_i = (x_1 \vee x_2 \vee \dots \vee x_k)$
- Each literal is a variable or a negation of a variable
- Each variable can appear multiple times in the expression.

For some constant k , the k -SAT problem requests if there is a satisfying assignment that makes a CNF formula $F = C_1 \wedge C_2 \wedge \dots \wedge C_m$ true. The problem of MAX k -SAT can be defined by specifying implicitly a pair (O, SC) where O is the set of all potentials solution $(\{0, 1\}^n)$ and SC is a mapping $O \rightarrow \mathbb{N}$ called score of the assignment that is the number of true clauses. Each solution is viewed as binary vector. Consequently, the problem consists to define the best binary vector that maximizes the number of true clauses in the Boolean formula.

Clearly, there are 2^n potential satisfying assignments for this problem. It has been proven that the MAX k -SAT problem is NP-complete for any $k \leq 3$. In this paper we deal with the Max 3-SAT problem. It is obviously that this problem is a combinatorial optimization problem. It appears to be impossible to obtain exact solutions in polynomial time. The main reason is that the required computation grows exponentially with the size of the problem. Therefore, it is often desirable to find near optimal solutions to these problems. Efficient heuristic algorithms offer a good alternative to accomplish this goal.

3 State-of-the-art SAT solving algorithms

Modern Max-SAT solvers have deeply improved the techniques and algorithms to find optimal solutions. In practice there are two broad classes of algorithms for solving instances of Max-SAT: Complete and Incomplete methods. The complete methods examine the entire search space, they fully answer for the problem of satisfiability of an instance [4, 23]. In the worst case, the computation time required for the execution of such methods increases exponentially with the size of the instance to solve. The Davis and Putnam method (DP) [24] is one of the first methods dedicated for solving the SAT problem. The DP method is based on the application of the resolution rule in order to eliminate

variables. Unfortunately, this mechanism may generate an exponential number of resolved clauses. The Davis, Putnam, Logemann and Loveland procedure (commonly known DPLL) is an enhancement of the algorithm DP [7]. The DPLL is a highly efficient and still forms the basis of the most effective SAT solvers. DPLL escapes the exponential clause generation and the high memory usage of DP. The DPLL takes advantage of two techniques that improve the selection of assignments to variables. One of these is Pure Literal Elimination [25]. A pure literal is a literal that occurs only in positive form or only in negative form in the CNF formula. The DPLL procedure assigns a value to the pure variable in a way to make all clauses containing them true which permits the algorithm to eliminate these clauses from the formula. The second technique used by the DPLL algorithm is Unit Propagation [26]. A clause is said to be a unit clause if contains only a single unassigned literal. The unit propagation algorithm consists in assigning the value to single unassigned literals contained in all the unit clauses that makes the clauses true. The unit propagation algorithm helps to accelerate the DPLL algorithm by recognizing assignments that must be made and therefore excluding certain parts of the search tree. Currently, the most of exact methods are based mainly on the DPLL procedure. The main difference between them is the branching heuristics used to explore the search tree [23, 27]. The fact that complete algorithms require huge amount of computing time, even for relatively small problem instances, incomplete methods have been developed to resolve SAT or MAX-SAT problems of large size in reasonable time. Although, incomplete methods do not guarantee finding the best solutions, because they do not explore the entire space of possible solutions, they are able to find quickly a good solution towards the cost and constraints of the problem. The incomplete methods are principally based on metaheuristics such as: local search methods, evolutionary algorithms or hybrid methods. The local search methods are widely used to solve Max 3-Sat problems [11]. Among the popular methods of this class, we can cite the popular method GSAT [28] and its improved variant Walksat [29]. GSAT starts with a random assignment and iteratively apply a set of flips by using a specific heuristics in order to enhance the number of satisfied clauses. Walksat is an evolution of GSAT in which the heuristics used to select the variable to flip is improved. The main difference lies in the estimated neighborhood size: Walksat reduces greatly the number of neighbors as it selects a false clause, and only one variable of this clause may be flipped. One biggest improvement to GSAT method is adopting the strategy of "random walk" [30] while solving the problem. This strategy allows us to accelerate the convergence and allow avoiding some local minima. The idea of this strategy is to choose a variable to flip among the variables involved in randomly selected false clause. Unfortunately, Walksat suffers from the local minima problem. To overcome this problem, Walksat insert some randomness in the search

process. New heuristics were integrated in Walksat to choose a variable of the clause to flip such as Random, Best Novelty, Rnovelty and Tabu [30]. Evolutionary algorithms have been applied to SAT problems. However, the use of a pure evolutionary algorithm has been unsuccessful in Max Sat problems. In fact, Rana and Whitley showed that a classical genetic algorithm is unsuitable for Max 3-Sat problem [16], because this problem requires more intensification search than diversification search. Therefore, hybridization between evolutionary algorithms and local search methods are needed to find success. This kind of hybridization is called memetic algorithms [31]. Recently, several memetic algorithms were proposed to deal with the Max Sat problems for example: FlipGA [9] based on hybrid genetic algorithm and a special flip heuristics; GASAT [8] based on hybrid genetic algorithm and tabu search and QGASAT [32] based on quantum evolutionary algorithm and an adapted quantum local search procedure. Finally, a new kind of hybridization based on incomplete and complete algorithms has emerged. Although the exact methods are able to provide an optimal solution for instances of small size, incomplete methods are usually able to find solutions close to optimum for instances of large sizes. It is therefore natural that, starting from this observation, many hybrid methods of these two resolution paradigms were considered. Hybridization consists to use heuristics inspired DPLL for local search heuristics, and reciprocally the local search methods for DPLL algorithm. In such combinations, the incomplete methods are used to quickly identify promising areas of search space in which the exact methods practiced an exhaustive exploration. There are several methods in the literature based on this kind of hybridization [33]

4 An Overview of Quantum Computing

Quantum Computing (QC) is an emergent field calling upon several specialties: physics, engineering, chemistry, computer science and mathematics [17]. QC uses the specificities of quantum mechanics for the processing and the transformation of information. The aim of this integration of knowledge is the realization of a quantum computer in order to carry out certain calculations much more quickly than with a traditional computer. This acceleration is made possible while benefiting from the quantum phenomena such as the superposition of states, the entanglement and the interference. A particle according to principles of quantum mechanics can be in a superposition of states. By taking account of this idea, one can define a quantum bit or the qubit which can take value 0, 1 or a superposition of the two at the same time. Its state can be given by:

$$\Phi = a|0\rangle + b|1\rangle \quad (1)$$

Where $|0\rangle$ and $|1\rangle$ represent the classical bit values 0 and 1 respectively; a and b are complex numbers such that:

$$|a|^2 + |b|^2 = 1 \quad (2)$$

The probability that the qubit collapses towards 1 (0) is $|a|^2$ ($|b|^2$). This idea of superposition makes it possible to represent an exponential set of states with a small number of qubits. According to the quantum laws like interference, the linearity of quantum operations and entanglement make the quantum computing more powerful than the classical machines. Each quantum operation will deal with all the states present within the superposition in parallel. For in-depth theoretical insights on quantum information theory, one can refer to [17, 18]. A quantum algorithm consists in applying of a succession of quantum operations on quantum systems. Quantum operations are performed using quantum gates and quantum circuits. Yet, a powerful quantum machine is still under construction. By the time when a powerful quantum machine would be constructed, researches are conducted to get benefit from the quantum computing field. Since the late 1990s, merging quantum computation and evolutionary computation has been proven to be a productive issue when probing complex problems. The purpose of this combination is to increase the profit of each one of these two areas by reciprocally inspiring each from the other. Like any other evolutionary algorithm, a Quantum Genetic Algorithm QGA relies on the representation of the individual, the evaluation function and the population dynamics [19]. The particularity of QGA 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. Quantum genetic algorithms were proposed to solve many difficult combinatorial optimization problems, and the experimental results demonstrated that these algorithms were far more superior to conventional GAs [19–22]

5 The Proposed Approach

Exploitation of the optimization properties of quantum computing is an attractive way to probe complex problems like the MAX 3-SAT problem. Within this perspective, we are interested in applying quantum computing principles improved by new stochastic local search method to solve this problem. The first quantum inspired genetic algorithm for Max 3-SAT was proposed in [32]. The authors have integrated an adapted flip procedure in the core of the quantum genetic algorithm. However, the behavior study of this hybrid algorithm showed that the convergence speed is slow and the performance results are not competitive compared to other recent techniques. For this purpose, we propose in

this paper a new local search operator. This latter is an adapted divided and conquer algorithm based on DPLL procedure. The development of the suggested approach called QGADPLL is based mainly on a quantum representation of the search space associated with the problem and a quantum dynamic used to explore this space by operating on the quantum representation by using quantum operations. The quantum dynamic is enhanced by a local search procedure based on DPLL algorithm. 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 Boolean assignments and the definition of quantum evolutionary operators. Then, we describe how these defined concepts have been combined with local search algorithm to deal with the Max 3-Sat problem.

5.1 Quantum representation of MAX 3-SAT solution

To successfully apply quantum computing principles on Max 3-Sat problem, we have needed to map potential solutions into a quantum representation that could be easily manipulated by quantum inspired evolutionary operators. The Boolean assignment is represented as binary vector of size n . In terms of quantum computing, each solution is represented as a quantum register of size n where each bit is associated with one variable as shown in figure 1. The register contains superposition of all possible solutions. Each column $\begin{pmatrix} a_i \\ b_i \end{pmatrix}$ 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 Equation (1). For each qubit, a binary value is computed according to its probabilities $|a|^2$ and $|b|^2$. $|a|^2$ and $|b|^2$ are interpreted as the probabilities to have respectively 0 or 1. Consequently, all feasible solution can be represented by a Quantum Vector QV that contains the superposition of all possible solutions. This quantum vector can be viewed as a probabilistic representation of all the MAX 3-SAT solutions. The figure 2 shows an example of equi-probable quantum chromosome (we have the same probability to get 0 or 1 by the measurement). When embedded within an evolutionary framework, it plays the role of the chromosome. Only one chromosome is needed to represent the entire population.

5.2 Quantum operators

The quantum operations which are the basis of the quantum genetic dynamics are as follows:

$$\left(\begin{array}{c|c|c|c} \mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_m \\ \mathbf{b}_1 & \mathbf{b}_2 & \dots & \mathbf{b}_m \end{array} \right)$$

Fig. 1: Quantum representation of the SAT solution

$$\left(\begin{array}{c|c|c|c|c} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{array} \right)$$

Fig. 2: Example of Equiprobable quantum solution of Max 3 – Sat problem

$$\left(\begin{array}{c|c|c|c} \mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_n \\ \mathbf{b}_1 & \mathbf{b}_2 & \dots & \mathbf{b}_m \end{array} \right) \xrightarrow{\text{Measure}} (0,1,\dots,1)$$

Fig. 3: Measure operation

5.2.1 Measurement

This operation transforms by projection the quantum matrix into a binary matrix as shown in Figure 3. 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 its probabilities and . 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 probability is greater than Pr, else the bit value is 0. Moreover, 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.

5.2.2 Quantum interference

The operation of interference is useful to intensify search around the best solution. 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. This operation can be accomplished by using a unit transformation which achieves a rotation whose angle is a

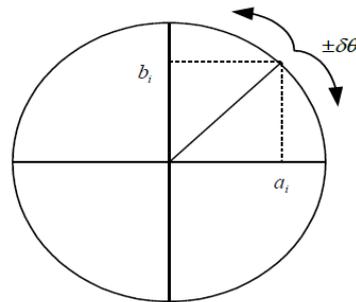


Fig. 4: Quantum interference

function of the amplitudes a_i , b_i and of the value of the corresponding bit in the solution reference. The Figure 4 illustrates a quantum interference operation. The values of the rotation angle θ is chosen so that 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 elements value in the binary vector (table1).

Table 1. Lookup table of the rotation angle

a	b	Reference bit value	Angle
> 0	> 0	1	$+\delta\theta$
> 0	> 0	0	$-\delta\theta$
> 0	< 0	1	$-\delta\theta$
> 0	< 0	0	$+\delta\theta$
< 0	> 0	1	$-\delta\theta$
< 0	> 0	0	$+\delta\theta$
< 0	< 0	1	$+\delta\theta$
< 0	< 0	0	$-\delta\theta$

5.2.3 Mutation operator

This operator allows exploring new solutions and thus enhances the diversification capabilities of the search process. It allows moving from the current solution to one of its neighbors. It consists of choosing a random qubit with a defined probability P_m , and then there is a permutation between its components a_i and b_i as shown on the figure 5, that will inverse the probabilities of having the values 0 and 1 when applying a measurement.

5.2.4 Crossover operators

Crossovers are important for promoting the exchange of high quality blocks within the population. The quantum

$$\begin{array}{c} \left(\begin{array}{cc|cc} 0.44 & 0.77 & \mathbf{0.44} & 0.00 & 1.00 \\ 0.99 & 0.77 & -\mathbf{0.99} & 1.00 & 0.00 \end{array} \right) \\ \Downarrow \\ \left(\begin{array}{cc|cc} 0.44 & 0.77 & -\mathbf{0.99} & 0.00 & 1.00 \\ 0.99 & 0.77 & \mathbf{0.44} & 1.00 & 0.00 \end{array} \right) \end{array}$$

Fig. 5: Quantum Mutation

$$\begin{array}{c} \left(\begin{array}{cc|ccc} 0.9887 & 0.4132 & 0.7955 & 0.8733 & 0.7676 \\ 0.0224 & 0.8293 & 0.3672 & 0.2373 & 0.4108 \end{array} \right) \\ \Updownarrow \\ \left(\begin{array}{cc|ccc} 0.3757 & 0.3368 & \mathbf{0.7394} & \mathbf{0.1644} & \mathbf{0.4073} \\ 0.8588 & 0.8866 & \mathbf{0.4534} & \mathbf{0.9730} & \mathbf{0.8341} \end{array} \right) \\ \Downarrow \\ \left(\begin{array}{cc|ccc} 0.9887 & 0.4132 & \mathbf{0.7394} & \mathbf{0.1644} & \mathbf{0.4073} \\ 0.0224 & 0.8293 & \mathbf{0.4534} & \mathbf{0.9730} & \mathbf{0.8341} \end{array} \right) \\ \left(\begin{array}{cc|ccc} 0.3757 & 0.3368 & 0.7955 & 0.8733 & 0.7676 \\ 0.8588 & 0.8866 & 0.3672 & 0.2373 & 0.4108 \end{array} \right) \end{array}$$

Fig. 6: Quantum crossover

crossovers have the same principle as the crossovers in conventional genetic algorithms. However, it operates on quantum chromosomes. The chromosomes are chosen randomly. Then, the two selected chromosomes exchange some selected quantum qubits between them. The advantage of quantum crossover is the exploration of new solutions. The crossover is very important in order to avoid to be trapped in local optima, because all quantum chromosomes evolve to the same direction. For example the figure 6 shows an example of quantum crossover. It should be noted that is better to use specific crossover operations more adapted for the max sat problems like those used in GASAT [8]. View that the quantum representation offers a great diversity; it is preferably to use small values for the probabilities of mutation and crossover in order to keep good performance of GQA.

5.3 The Local Search-based DPLL

The study of the performance of the method QGASAT based on a quantum genetic algorithm with a simple local

search technique [32] showed that the performance to find the exact solutions for satisfiable instances is low and the convergence speed is slowed down. In this section, we show how a quantum genetic algorithm can be greatly improved by hybridizing with a new local search based on a complete method. The complete methods give optimal solutions because it performs an exhaustive search and thus fully exploits the search space. Unfortunately, they suffer from the explosion combinatorial problem. On other hand, incomplete methods are characterized by great simplicity and speed as they explore certain areas of search space. They however, give less optimal solutions. It is interesting to merge complete and incomplete methods in order to take advantage of the best features of both approaches. Our idea is to integrate the intensification abilities of the exact method DPLL in the quantum genetic algorithm that has the diversification faculties. The goal of this hybridization is to obtain a good compromise between these two basic strategies of search and without increasing the computing time. That is while, the exact algorithm DPLL is incorporated into the algorithm QGADPLL as a local search method (Figure 7). The idea is based on a well-known concept in computer science: "divide and conquer" which consists to divide arbitrarily the problem into several sub-problems easily resolvable. The partial solutions found are inserted in the solution of the original problem. Consequently, to use the efficiency of exact algorithm without increasing the execution time, the new local search method applies iteratively the DPLL algorithm on small parts of the Boolean formula. The principle is to first select a continuous subset of clauses in the Boolean formula (Figure 8). Therefore, this chosen subset of clauses constitutes a Boolean formula with few variables and few clauses, which makes the application of the exact algorithm easy. In our method, the size of window selection is dynamic in order to maintain greater efficiency. Our local search method is generic which can be integrated easily in different metaheuristics. The general scheme of this procedure is given by the bellow algorithm (Figure 9). The Figure 8 gives an example of how apply our local search based DPLL. For example, we have a CNF formula of 9 clauses and 13 Boolean variables. The first step consists in selecting a set of random clauses (IC). Consequently, the obtained formula F' contains a few clauses (5 clauses) and a few variables (for example 6 variables). After the application of the DPLL algorithm on the formula F' we get a partial exact assignment for the 6 variables of the formula F' . Finally, the variables values involved in the chosen clauses are replaced by the new ones found by the DPLL procedure.

5.4 Outline of the proposed framework

Now, we describe how the quantum representation, the quantum operators and the local search based DPLL have been embedded within an evolutionary algorithm and

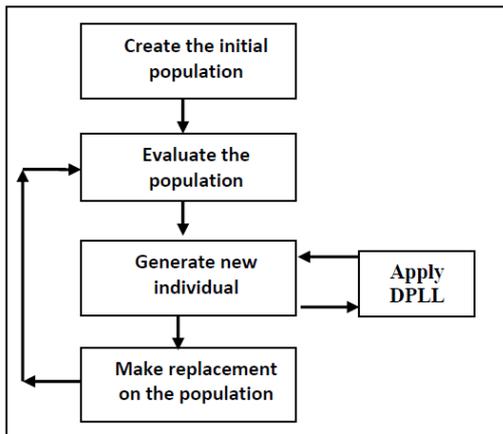


Fig. 7: General Scheme of a hybrid QGA with an exact method

Algorithm 1: The local search based DPLL
input: a quantum chromosome VQ, Boolean formula ϕ in CNF, Nbtries.
output: boolean assignment VB

```

Begin
VB=measure(VQ) ;
Evaluate(VB) ;
Repeat
  Select randomly a subset of clauses
  IC;
  Apply the DPLL procedure on IC;
  Modify the best variable involved by
  according to the result given by DPLL
  procedure;
  Evaluate(VB) ;
  if (there is improvement) then
    Save VB;
  else
    undo the changes made to VB;
  endif
until Nbtries ;
End
  
```

Fig. 9: The local search based DPLL

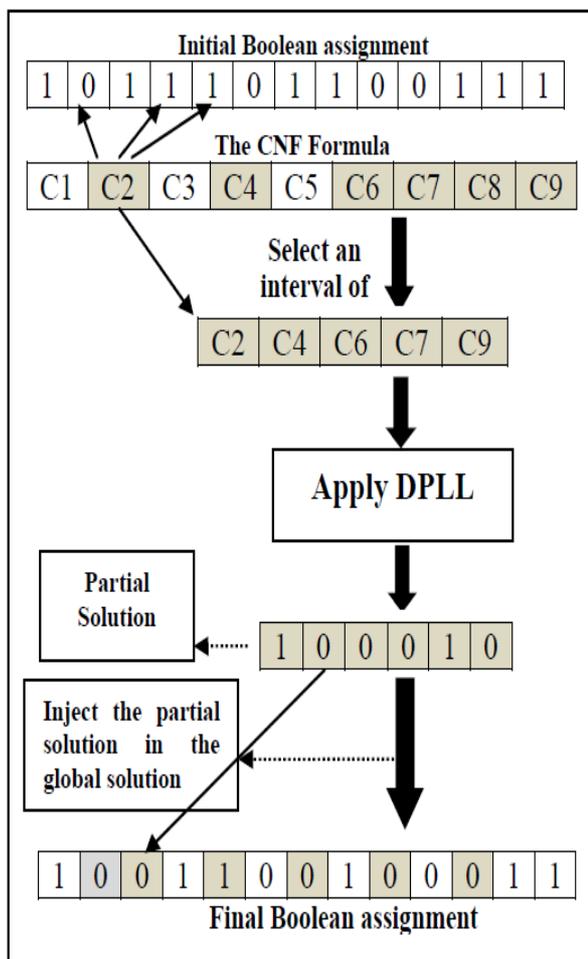


Fig. 8: An Example of partial application of the local search based DPLL

resulted in a hybrid stochastic algorithm performing variable truth assignment search. In more details, the proposed QGADPLL can be described as in the algorithm 2. Given a Boolean formula in CNF, first, a population of quantum chromosomes is randomly generated to represent all possible variable assignments. Each quantum chromosome contains random numbers between the numbers -1 and 1. Secondly, the initial population is evaluated using the fitness function. The evaluation of each solution is preceded by the measure operation in order to get a binary solution. The objective function to be maximized in the Max 3-Sat problem is the number of satisfied clauses in the Boolean formula. However, its preferable to use an adaptive fitness function which help to center on difficult clauses and to escape from local optima. During the whole process we keep in memory the global best solution. The algorithm progresses through a number of generations according to a quantum based dynamics. At each iteration, the following main tasks are performed: the application of the crossover and mutation operations, the application of the local search method based DPLL, and the application of the interference operation. Finally, we apply the selection and replacement operators to generate the next population. The selection operation selects p quantum chromosomes among the quantum chromosomes representing the current population. It takes place in two steps: first a measurement operation is performed to extract the binary chromosomes representing the potential solutions. Afterwards, we select the p chromosomes from which derive the p best results and we generate also randomly new quantum chromosomes in order to keep a good

Algorithm 2: QGADPLL scheme
Input: Boolean formula in CNF form A
Output: best variable assignment, number of satisfied clauses

- (1) Construct randomly the initial population.
- (2) Evaluate the initial population.

Repeat

- (3) Apply the crossover operation.
- (4) Apply the mutation operation according to the permutation probability p_m .
- (5) Apply iteratively the local search procedure. (algorithm 1)
- (6) Evaluate the new population.
- (7) Apply selection and replacement operators.
- (8) Apply the interference operation according to the best solution.

Until a termination-criterion is reached.

Fig. 10: QGADPLL scheme

diversity. The whole process is repeated until reaching a stopping criterion (Figure 10).

6 Implementation and Evaluation

QGADPLL is implemented in Java and tested on a micro-computer 3 GHz and 1 GB of memory. We have used the package SAT4J [34] which contains a java implementation of an exact method based DPLL. In order to assess the efficiency and accuracy of our approach, we have performed several tests taken from the AIM benchmark instances [35]. The AIM instances are all generated with a particular Random-3-SAT instance generator. The used AIM instances include yes-instances with low and high *clause/variable* ratios that have exactly one solution. In our experiment, we have used only the satisfiable benchmarks because we want to evaluate the performances of our approach to find the exact solutions for yes-instances. We have compared our algorithm against three popular stochastic local search methods: Walksat, Novelty [36] and an Iterated Robust Tabu Search IROTS [37]; and two memetic algorithms: Clonsat [38] which is based on Clonal selection algorithm and walksat and QGASAT [32]. The purpose of the last comparison is to show the impact of local search on quantum genetic optimization in order to solve the MAX 3-SAT problem. In all experiments, the size of population is 4 and the selection size is $p=3$. The permutation probability is a tunable parameter which was set to 0.15, the crossover probability is 0.6, the interference angle is $p/20$, and the iteration number is 100 iterations for quantum genetic algorithm and 30 iterations for the local search procedure. The size of the window selection in the local search based DPLL is chosen between 15 and 25 clauses for small instances and between 20 and 80 clauses for large instances. Finally, statistical tests of Friedman

were carried out to test the significance of the difference in the accuracy of each method in this experiment. The table two summarizes the obtained results. For each program, we have taken the best of three consecutive runs. The results found seem to be very promising and demonstrate the feasibility and efficiency of our approach. Indeed, QGADPLL is able to find the exact solutions for all the instances. According to the Friedman test, our method is better than QGASAT which confirms that the local search method used in QGASAT is not good to deal with yes-instances (Figure 11). The results show that the integration of the new local search based exact procedure in the quantum genetic algorithm provides a significant improvement in the performance and efficiency of our approach compared to the method QGASAT. Moreover, our approach needs less iteration to find good solutions. The results of the programs IROTS and Clonsat are very close to the exact solutions. However the two local search methods Walksat and Novelty are not good in this experiment. The study of the optimization behavior of our approach reveals that finding the optimal solution is not conducted in a random manner. The best solution is gradually improved until the terminal conditions are reached. Figure 12 shows an example of the variation of the best QGADPLL solution found during the optimization process of three consecutive executions. The effectiveness of our approach is explained by the good combination between diversification and intensification which leads the algorithm to effectively explore the search space and locate a good solution. In addition, our idea is a very promising approach to solve satisfiable instances of large size. Finally, the performance of QGA without the local search algorithm is rather poor.

Table 2. Results

Test	#var	#cls	Novelty	Wsat	QGASAT	QGADPLL	IROTS	Clonsat
aim-50-1.6-yes1	50	80	79	79	79	80	79	79
aim-50-1.6-yes2	50	80	79	79	79	80	79	80
aim-50-1.6-yes3	50	80	79	79	79	80	80	80
aim-50-1.6-yes4	50	80	79	79	79	80	80	80
aim-50-2.0-yes1	50	100	99	99	99	100	100	100
aim-50-2.0-yes2	50	100	99	100	100	100	100	100
aim-50-2.0-yes3	50	100	99	99	99	100	100	100
aim-50-2.0-yes4	50	100	99	100	100	100	100	100
aim-50-3.4-yes1	50	170	170	170	170	170	170	170
aim-50-3.4-yes2	50	170	170	170	170	170	170	170
aim-50-3.4-yes3	50	170	170	170	170	170	170	170
aim-50-3.4-yes4	50	170	170	170	170	170	170	170
aim-50-6.0-yes1	50	300	300	300	300	300	300	300
aim-50-6.0-yes2	50	300	300	300	300	300	300	300
aim-50-6.0-yes3	50	300	300	300	300	300	300	300
aim-50-6.0-yes4	50	300	300	300	300	300	300	300
aim-100-1.6-yes1	100	160	159	159	159	160	159	159
aim-100-1.6-yes2	100	160	159	159	159	160	159	159
aim-100-1.6-yes3	100	160	159	159	159	160	159	159
aim-100-1.6-yes4	100	160	159	159	159	160	159	159
aim-100-2.0-yes1	100	200	199	199	199	200	199	199
aim-100-2.0-yes2	100	200	199	199	199	200	199	199
aim-100-2.0-yes3	100	200	199	199	199	200	199	199
aim-100-2.0-yes4	100	200	199	199	199	200	199	199
aim-100-3.4-yes1	100	340	340	340	339	340	340	340
aim-100-3.4-yes2	100	340	340	340	337	340	340	340
aim-100-3.4-yes3	100	340	340	340	340	340	340	340
aim-100-3.4-yes4	100	340	340	340	340	340	340	340
aim-100-6.0-yes1	100	600	600	600	600	600	600	600
aim-100-6.0-yes2	100	600	600	600	600	600	600	600
aim-100-6.0-yes3	100	600	600	600	600	600	600	600
aim-100-6.0-yes4	100	600	600	600	600	600	600	600
aim-200-2.0-yes1	200	400	399	399	399	400	399	399
aim-200-2.0-yes2	200	400	399	399	398	400	399	399
aim-200-2.0-yes3	200	400	399	399	399	400	399	399
aim-200-2.0-yes4	200	400	399	399	399	399	399	399
aim-200-6.0-yes1	200	1200	1200	1200	1200	1200	1200	1200
aim-200-6.0-yes2	200	1200	1200	1200	1200	1200	1200	1200
aim-200-6.0-yes3	200	1200	1200	1200	1200	1200	1200	1200
aim-200-6.0-yes4	200	1200	1200	1200	1200	1200	1200	1200

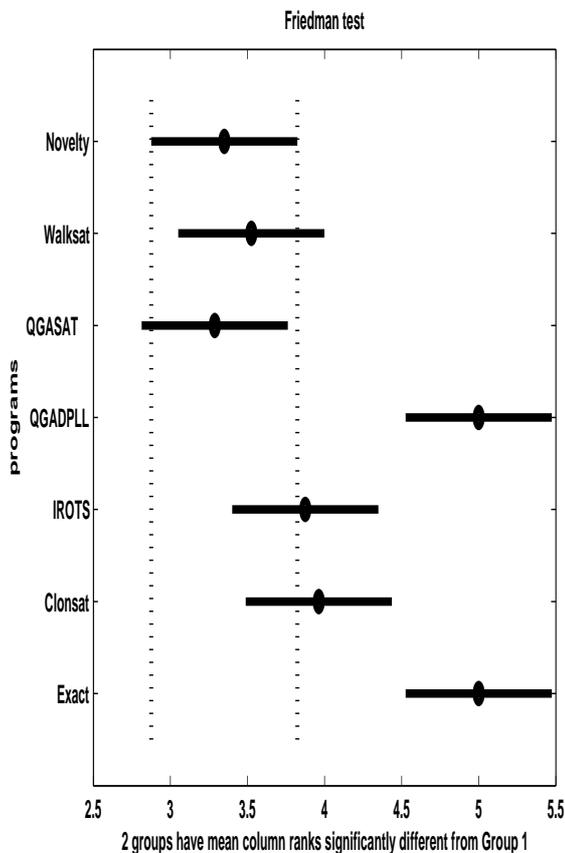


Fig. 11: Freidman test compares the methods QGADPLL, Walksat, Novelty, IROTS, QGSAT and Clonsat against exact method. Dash lines: not successful programs. Solid line: successful program

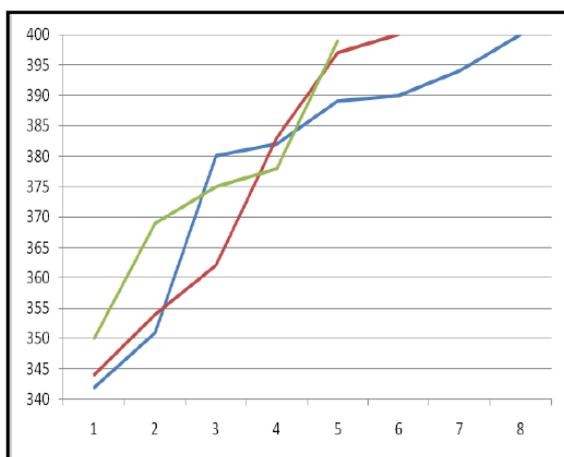


Fig. 12: The behavior of the best solution for the test : *aim - 200 - 2.0 - yes1 - 1*

7 conclusion

In this paper, we have proposed a new approach called QGADPLL to solve the problem MAX 3-SAT. QGADPLL combines strengths of both DPLL and Quantum Genetic Algorithms. The quantum representation of the solutions allows the coding of all the potential solution with a certain probability. The optimization process consists of the application of a quantum dynamics constituted of quantum operations such as the interference, the quantum mutation and measurement, enhanced by local search procedure. The choice of the local search procedure is crucial for the effectiveness of the resulting algorithm. There is a clear potential of using the DPLL search as local search procedure. Moreover, our approach is flexible, so we can use other stochastic local search algorithms. The size of the population is considerably reduced thanks to the superposition principle. The experimental study proves the feasibility and the effectiveness of our approach. As ongoing work we study the effect of the parallelization on the performance of our approach. There are several issues to improve our approach. For example, we can divide a large instance into several sub-problems, and then run in parallel the DPLL procedure on all these sub-problems. Another improvement is to identify and exploit the hidden structure in a SAT problem, which helps to counteract the combinatorial explosion of resolution.

References

- [1] Cook, S.A. The Complexity of Theorem Proving Procedures. In: Proc. 3rd Ann. ACM Symp. On Theory of Computing, Association for Computing Machinery, 151-158 (1971).
- [2] Biere A., Cimatti, A., Clarke E., and Zhu, Y. Symbolic model checking without BDDs. In Proc. of the Workshop on Tools and Algorithms for the Construction and Analysis of Systems (TACAS99), LNCS, Springer-Verlag, **1579**, 19-207 (1999).
- [3] Bouhmala, N. and Granmo, O. Solving Graph Coloring Problems Using Learning Automata. In Proceedings of the Eighth European Conference on Evolutionary Computation in Combinatorial Optimisation, Lecture Notes in Computer Science, **4972**, 277-288 (2008).
- [4] Borchers, B., Furman, J. Two-Phase Exact Algorithm for MAX-SAT and Weighted MAX-SAT Problems, Journal of Combinatorial Optimization, **2**, 299- 306 (1999).
- [5] Menai, M., Batouche, M. A Backbone-Based Co-evolutionary Heuristic for Partial MAX-SAT, Artificial Evolution, 155-166 (2005).
- [6] Zhang, H., Shen, H. Exact Algorithms for MAX-SAT. Electronic Notes in Theoretical Computer Science, **86**, (2003).
- [7] Davis, M., Putnam, G., Loveland, D. A machine program for theorem proving. communication of the ACM, 94-397 (1962).
- [8] Lardeux, F., Saubion, F., Hao, J.K. GASAT: A genetic local search algorithm for the satisfiability problem. Evolutionary Computation, **14**, 223-253 (2006).

- [9] Marchiori, E., Rossi, C. A Flipping Genetic Algorithm for Hard 3-SAT Problems. In: Proc. of the Genetic and Evolutionary Computation Conference, **1**, 393-400 (1999).
- [10] Marques-Silva, J. P., Sakallah, K. A. GRASP: A Search Algorithm for Propositional Satisfiability. *IEEE Transactions on Computers*, **48**, 506-521 (1999).
- [11] Holger, H., Stutzle, T. Local search algorithms for SAT: An empirical evaluation. *Journal of Automated Reasoning*, **24**, 421-481 (2000).
- [12] Mautor, T., Michelon, P. MIMAUSA: A new hybrid scheme combining exact and local search. In Proceedings of the 2nd International Conference on Metaheuristics, Sophia-Antipolis, France, (1997).
- [13] Kroc, L., Saharawi, A., Gomes, C. P., Selman, B. Integrating systematic and local search paradigms: A new strategy for MaxSAT. *IJCAI*, 544-551 (2009).
- [14] Goldberg, D. *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, (1989).
- [15] Man, K. F., Tang, K. S. and Kwong, S. *Genetic Algorithms: Concepts and Designs*, Springer-Verlag, (1999).
- [16] Rana, S., Whitley, D., Heckendorn, R.B. A Tractable Walsh Analysis of SAT and its Implications for Genetic Algorithms. In proceedings of the 15th National Conference on Artificial Intelligence, 392-397 (1998).
- [17] Jaeger, G. *Quantum Information: An Overview*. Berlin: Springer, (2006).
- [18] Preskill, J. *Quantum Information and Computation*. Lecture Notes for Physics, California institute of technology, **229**, 1-321 (1998).
- [19] Han, K. H., Kim, J. H. Quantum-inspired evolutionary algorithm for a class of combinatorial optimization, *IEEE Trans. Evolutionary Computation*, **6**, 580-593 (2002).
- [20] Layeb, A. and Saidouni, D. E. Quantum Genetic Algorithm for Binary Decision Diagram Ordering Problem. *The International Journal of Computer Science and Network Security*, **7**, 130-135 (2007).
- [21] Layeb, A. A novel quantum inspired cuckoo search for knapsack problems. *International Journal of Bio-Inspired Computation*, **3**, 297-305 (2011).
- [22] Draa A., Meshoul S., Talbi H., Batouche M. A Quantum-Inspired Differential Evolution Algorithm for Solving the N-Queens Problem. *Int. Arab J. Inf. Technol.*, **7**, 21-27 (2010).
- [23] Matthew, W. M., Conor, F. M., Ying, Z., Lintao, Z., Sharad, M. Chaff: Engineering an Efficient SAT Solver. In Proceedings of 12th International Conference on Computer Aided Verification, LNCS, **1855**, (2001).
- [24] Davis, M., Putnam, H. A computing procedure for quantification theory. *Journal of the ACM*, **7**, 201-215 (1960).
- [25] Dowling, W., Gallier, J. Linear-time algorithms for testing the satisfiability of propositional Horn formulae. *Journal of Logic Programming*, **1**, 267-284 (1984).
- [26] Li, C.M., Many, F., Planes, J. Exploiting unit propagation to compute lower bounds in branch and bound Max-SAT solvers. In: Proc. of the 11th Int. Conf. on Principles and Practice of Constraint Programming, CP'05, Sitges, Spain, in: LNCS, **3709**, 403-414 (2005).
- [27] Xu, L., Hutter, F., Hoos, H.H., and Leyton-Brown, K. SATzilla: portfolio-based algorithm Selection for SAT. *Journal of Artificial Intelligence Research*, **32**, 565-606 (2008).
- [28] Selman, B., Levesque, H., Mitchell, D. GSAT- A new method for solving hard satisfiability problems. In Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92), San Jose, Ca., 440-446 (1992).
- [29] Selman, B., Kautz, H., Cohen, B. Noise strategies for improving local search. In: Proceedings of AAAI'94. MIT Press, Cambridge, MA, 337-343 (1994).
- [30] Tompkins, D.A.D., Hoos, H.H. UBCSAT: An Implementation and Experimentation Environment for SLS Algorithms for SAT and MAX-SAT. In: Hoos HH and Mitchell DG (Eds.) SAT 2004, LNCS, **3542**, 306-320 (2005).
- [31] Moscato, P., and Cotta, C. A Gentle Introduction to Memetic Algorithms. *Operations Research & Management Science*, **57**, 105-144 (2005).
- [32] Layeb, A., Saidouni, D. A New Quantum Evolutionary Local Search Algorithm for MAX 3-SAT Problem. In Proceedings of the 3rd international Workshop on Hybrid Artificial intelligence Systems, LNAI, **5271**, 172-179 (2008).
- [33] Kroc, L., Saharawi, A., Gomes, C.P., Selman, B. Integrating systematic and local search paradigms: A new strategy for MaxSAT. *IJCAI*, 544-551 (2009).
- [34] Le Berre, D., Parrain, A. The SAT4J library, Release 2.2, System Description, *Journal on Satisfiability, Boolean Modeling and Computation*, **7**, 59-64 (2010).
- [35] Asahiro, Y., Iwama, K., Miyano, E. Random Generation of Test Instances with Controlled Attributes. In *Cliques, Coloring, and Satisfiability: The Second DIMACS Implementation Challenge*. DIMACS Series on Discr. Math. and Theor. Comp. Sci., **26**, 377-394 (1996).
- [36] McAllester, D., Selman, B. and Kautz, H. Evidence for invariants in local search, In Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97), 321-326 (1997).
- [37] Smyth, K., Hoos, H. H., Stutzle, T. Iterated Robust Tabu Search for MAX-SAT, In Canadian Conference on AI, Lecture Notes in Computer Science, **2671**, 129-144 (2003).
- [38] Layeb, A., Deneche, A. H., Meshoul, M. A New Artificial Immune System for Solving the Maximum Satisfiability Problem. In: N. Garcia-Pedrajas et al. (Eds.): IEA/AIE 2010, part II, Springer LNAI, **6097**, 136-142 (2010).



Abdesslem Layeb

is an Associate Professor in the Department of Computer Science at the University of Constantine, Algeria. He is a member of MISC Laboratory. He received his PhD in Computer Science from the University Mentouri

of Constantine, Algeria. Layeb is interested in the combinatorial optimization methods and their applications to solve several problems like bioinformatics, imagery, routing problems, etc.

Djamel eddine Saidouni was born in Algeria in 1968. He obtained his PhD degree from Paul Sabatier university, France, in 1996. Djamel Eddine is interested to the following topics : maximality semantics , formal methods , real time system , state explosion problem , models for concurrency , refinement. Dr Djamel Eddine is currently member of the RT-LOTOS project and the author of the true concurrency model for process algebra called Maximality Labeled Transition system. Dr Djamel Eddine has many publications in theoretical computer science and formal methods . Dr Djamel Eddine is currently a professor in the department of computer science at the University of Constantine. He is also the head of the research group on formal methods.