Hybrid Quantum Scatter Search Algorithm for Combinatorial Optimization Problems

Abdesslem Layeb
Computer Science Department,
University Mentouri of Constantine, Algeria

ABSTRACT. This paper presents a new evolutionary algorithm called Quantum Inspired Scatter Search Algorithm (QISSA). This one is new framework relying on Quantum Computing principles and Scatter Search algorithm. The contribution consists in defining an appropriate representation scheme that allows applying successfully on combinatorial optimization problems some quantum computing principles like qubit representation and superposition of states. This representation scheme is embedded within a Scatter Search algorithm leading to an efficient hybrid framework which achieves better balance between exploration and exploitation capabilities of the search process. Experiments on two combinatorial problems show the effectiveness of the proposed framework and its ability to achieve good quality solutions.
KEYWORDS: Quantum Computing, Scatter Search, Binary Decision Diagram

Introduction

Metaheuristics are a family of optimization techniques inspired by nature used to solve difficult optimization problems often from the fields of operational research, engineering or artificial intelligence, for which there is no known method more effective. Scatter Search (SS) is one of these methods [MLG06, GLW00]. SS is a new optimization algorithm based population. In SS process, each solution is encoded with n-dimensional vector. The Scatter Search method uses a new combination strategy for combining solution vectors that have been showed effective in a variety of optimization problems. Scatter search operates on a set of solutions called the *reference set*, by combining these solutions to create new ones. Compared to the genetic algorithms, the size of population in scatter search is small.

Quantum computing is a new research field that encompasses investigations on quantum mechanical computers and quantum algorithms [Jae06]. 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 [HK04, TDB04, LS07]. The aim is to get benefit from quantum computing capabilities to enhance both efficiency and speed of classical evolutionary algorithms. Consequently, lots of quantum inspired algorithms have merged such as Quantum inspired Genetic Algorithms QGA [HK04, TDB04], A Quantum-Inspired Differential Evolution QDE [LS07, LS08] etc.

Within this issue, we propose in this paper a new hybrid algorithm called Quantum Inspired Scatter Search Algorithm (QISSA) to cope with combinatorial optimization problems. The proposed algorithm combines scatter 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 QISSA is the integration of the quantum operators in the scatter search dynamics in order to optimize a defined objective function. The QISSA distinguishes with other evolutionary algorithms in that it offers a large exploration of the search space through intensification and diversification. We have tested our algorithm on two NP-Complete problems: the Max 3-Sat problem [LST08] and the variable ordering problem of the binary decision diagram.

Consequently, the remainder of the paper is organized as follows: A brief introduction to quantum computing is presented in next section. The section 2 presents the QISSA framework. Some basis concepts of BDD are presented in section 3. The proposed approach is described in section 4. Section 5 illustrates some experimental results. Then, we finish by giving conclusion and some perspective.

1 An Overview of Quantum Computing

Quantum Computing (QC) is an emergent field calling upon several specialties: physics, engineering, chemistry, computer science and mathematics. 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 [GLW00]:

$$\Psi = \alpha\,|0\rangle + b|1\rangle \qquad (1)$$

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

$$|\alpha|^2 + |b|^2 = 1 \qquad (2)$$

The probability that the qubit collapses towards 1 (0) is $|\alpha|^2$ ($|b|^2$).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 [Jae06].

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. Like any other Evolution Algorithm, a Quantum inspired Evolution 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 [TDB04, LS08].

2 Quantum Inspired Scatter Search

Scatter Search (SS) is new algorithm of optimization [MLG06, GLW00]. It derives its foundations from strategies originally proposed for combining decision rules and constraints in the context of integer programming. In SS process, each solution is encoded with n-dimensional vector. The Scatter Search method uses strategies for combining solution vectors that have been showed effective in a variety of optimization problems. Scatter search operates on a set of solutions, the *reference set*, by combining these solutions to create new ones. Compared to the genetic algorithms, the size of Population in scatter search is small. SS basically consist of the following five methods:

1. A *Diversification Generation Method:* generation of starting set of solution.
2. An *Improvement Method:* transform the initial population into one or more enhanced trial solutions.
3. A *Reference Set Update Method:* to build and maintain a *reference set* consisting of the *b* "best" solutions found
4. A *Subset Generation Method:* to operate on the reference set, to produce a subset of its solutions as a basis for creating combined solutions.
5. A *Solution Combination Method:* to combine a given subset of solutions produced by the Subset Generation Method in order to produce new solutions. The combination method is analogous to the crossover operator in genetic algorithms although it should be capable of combining more than two solutions.

In this paper, we present a new hybrid evolutionary algorithm called Quantum Inspired Scatter Search (QISSA) which integers the quantum computing principles such as qubit representation, measure operation and quantum mutation, in the core the scatter search algorithm. The quantum principles which are the base of the quantum inspired scatter search algorithm are as follows.

2.1 Quantum representation

In order to easily apply quantum principles on combinatorial optimization problems, we need to map potential solutions into a quantum representation that could be easily manipulated by quantum operators. In practical terms, this representation keeps knowledge of the most promising regions of the search space, and uses it to speed up convergence. In terms of quantum computing, each problem solution is represented as a quantum register 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 $|a_i|^2 + |b_i|^2 = 1$. For each qubit, a binary value is computed according to its probabilities $|a_i|^2$, $|b_i|^2$ and some problem constraints. $|a_i|^2$ and $|b_i|^2$ are interpreted as the probabilities to have respectively 0 or 1. Consequently, all feasible solutions are present in a quantum register (fig.2) that contains the superposition of all possible solutions. This quantum presentation can be viewed as a probabilistic encoding of all potential solutions. When embedded within a scatter search framework, it increases the diversity of the scatter population.

$$\left(\begin{array}{c|c|c|c} \mathbf{a}_1 & \mathbf{a}_2 & \multirow{2}{*}{...} & \mathbf{a}_m \\ \mathbf{b}_1 & \mathbf{b}_2 & & \mathbf{b}_m \end{array}\right)$$

Figure 1 Quantum register

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

Figre 2 Quantum representation of variable ordering

2.2 Quantum operators

We have integrated in the scatter search framework, some of quantum operations. This integration helps to increase the optimization capacities of the scatter search.

2.2.1. Measurement: This operation transforms by projection the quantum vector into a binary vector (fig.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 $|a_i|^2$ and $|b_i|^2$. The binary matrix is then translated into a problem solution.

$$\begin{pmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \ldots & \mathbf{a}_n \\ \mathbf{b}_1 & \mathbf{b}_2 & & \mathbf{b}_m \end{pmatrix} \xrightarrow{Mesure} (0,1,\ldots,1)$$

Figure 3 Quantum measurement

2.2.2. The 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. 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 and of the value of the corresponding bit in the solution reference (fig 4). The values of the rotation angle δθ is chosen so that to avoid premature convergence. It 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 matrix (table 1).

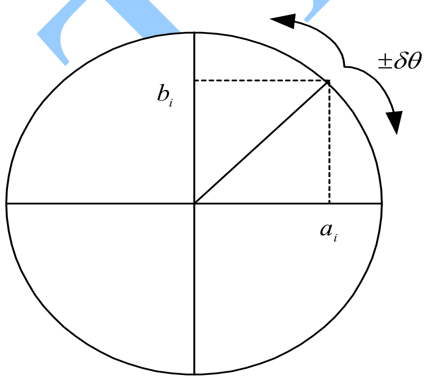

Figure 4. Quantum interference

Table 1.
Lookup table of the rotation angle

A	b	Reference bit value	Angle
> 0	> 0	1	+δθ
> 0	> 0	0	-δθ
> 0	< 0	1	-δθ
> 0	< 0	0	+δθ
< 0	> 0	1	-δθ
< 0	> 0	0	+δθ
< 0	< 0	1	+δθ
< 0	< 0	0	-δθ

2.2.3 Mutation operator: 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. We distinguish two types of quantum mutation:

- *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 5).

$$\begin{pmatrix} 0.44 & 0.77 & 0.44 & 0.00 & 1.00 \\ 0.99 & 0.77 & -0.99 & 1.00 & 0.00 \end{pmatrix}$$

Exchange

Figure 5. Inter-qubit quantum mutation

- *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's shown on the following figure.

$$\begin{pmatrix} 0.44 & 0.77 & 0.44 & 0.00 & 1.00 \\ 0.99 & 0.77 & -0.99 & 1.00 & 0.00 \end{pmatrix}$$

$$\Downarrow$$

$$\begin{pmatrix} 0.44 & 0.77 & -0.99 & 0.00 & 1.00 \\ 0.99 & 0.77 & 0.44 & 1.00 & 0.00 \end{pmatrix}$$

Figure 6. Intra-qubit quantum mutation

3 Binary decision diagram ordering problem

The representation by the Binary Decision Diagrams BDD [CGL94] is among the most known symbolic notations. The BDD is a data structure used to represent Boolean functions. The BDD is largely used in several fields since they offer a canonical representation and an easy manipulation. However, the BDD size depends on the selected variable order. Therefore it is important to find variable order which minimizes the number of nodes in a BDD. Unfortunately, this task is not easy considering the fact that there is an exponential number of possible variable ordering. Indeed, the problem of variable ordering was shown Np-hard [DB98]. For that, several methods were proposed to find the best BDD variable order and which can be classified in two categories. The first class tries to extract the good order by inspecting the logical circuits [Bry92], whereas, the second class is based on the dynamic optimization of a given order [BW96].

Mathematically, a Binary Decision Diagram is data structure used for representation of Boolean functions in the form of rooted directed acyclic graph. A BDD is a rooted directed acyclic graph $G = (V, E)$ with node set V containing two kinds of nodes, *non-terminal* and *terminal* nodes (fig.10). A non-terminal node v has as tag a variable $index(v) \in \{x_1, x_2, \ldots\ldots, x_n\}$ and two children $low(v)$, $high(v) \in V$. The final nodes are called *0-final and 1-final*. A BDD can be used to compute a Boolean function $f(x_1, x_2, .., x_n)$ in the following way. Each input $a = (a_1, a_2, \ldots\ldots, a_n) \in \{0,1\}^n$ defines a computation path through the BDD that starts at the root. If the path reaches a non-terminal node v that is labelled by x_i, it follows the path $low(v)$ if $a_i = 0$, and it follows the path $high(v)$ if $a_i = 1$. The label of the terminal node determines the return value of the BDD on input a. the BDD is called "ordered" if the different variables appear in the same order on all the ways from the root (fig. 7). It is important to note that for a given order of variables, the minimal binary decision graph is single. A BDD can be reduced while using the two following rules [DB98, FOH93, FFK88, CDM00]:

- Recognize and share identical sub-trees.
- Erase nodes whose left and right child nodes are identical.

It is very important to take into account the order of variables to be used when using the BDD in practice. The size of a BDD is largely affected by the choice of the variable ordering (fig. 8). Unfortunately, there are an exponential number of possible orders (permutation). It is completely clear that the problem of variables ordering is NP-difficult. The use of heuristics is essential to find acceptable solutions within reasonable times. Within this perspective, we are interested in applying quantum computing principles to solve the variable ordering problem.

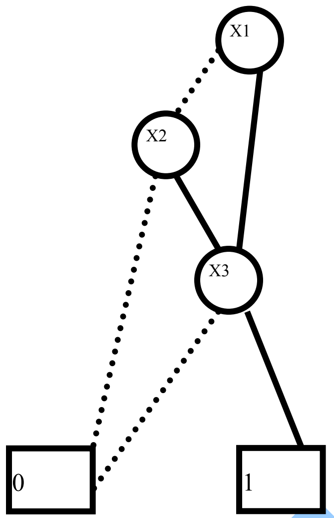

Figure 7. Binary Decision Diagram for the Boolean function:
$f = X1\,X3 + X2\,X3$

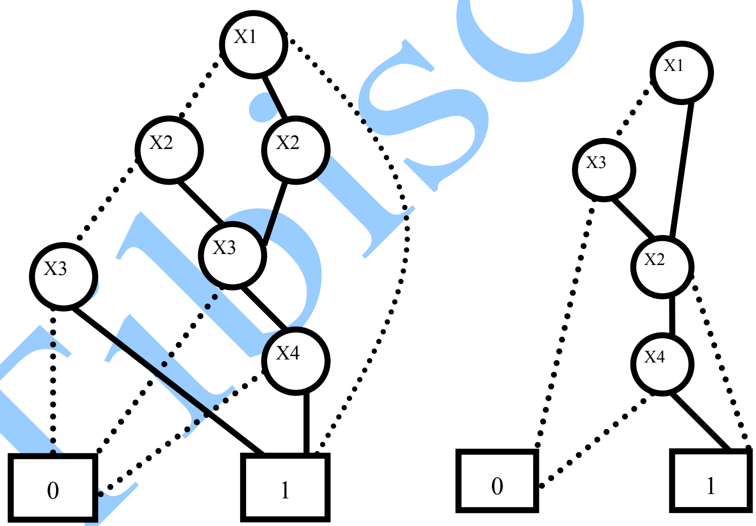

Figure 8.Two BDDs representation of the function $(x1 \vee x3) \wedge (x2 \Rightarrow x4)$, in the left the order of variable is: x1, x2, x3, x4; right: order: x1, x3, x2, x4

4 The QISSA for binary decision diagram ordering problem

The development of an approach based QISSA for an optimization problem called is based on a quantum representation of the research space associated with the problem and a quantum scatter search dynamics used to explore this space by operating on the quantum representation by using quantum operations and scatter search dynamics (fig. 9). Now, we describe how the representation scheme including quantum representation and quantum operators has been embedded within a scatter search algorithm and resulted in a hybrid stochastic algorithm performing variable order search.

Given a set *S* of BDD variables to be ordered, first, initial solutions are encoded in N chromosomes representing the initial population *PQM*. The algorithm progresses through a number of generations according to a quantum scatter search dynamics. Our algorithm basically consists of the following five steps of the basic SS algorithm.

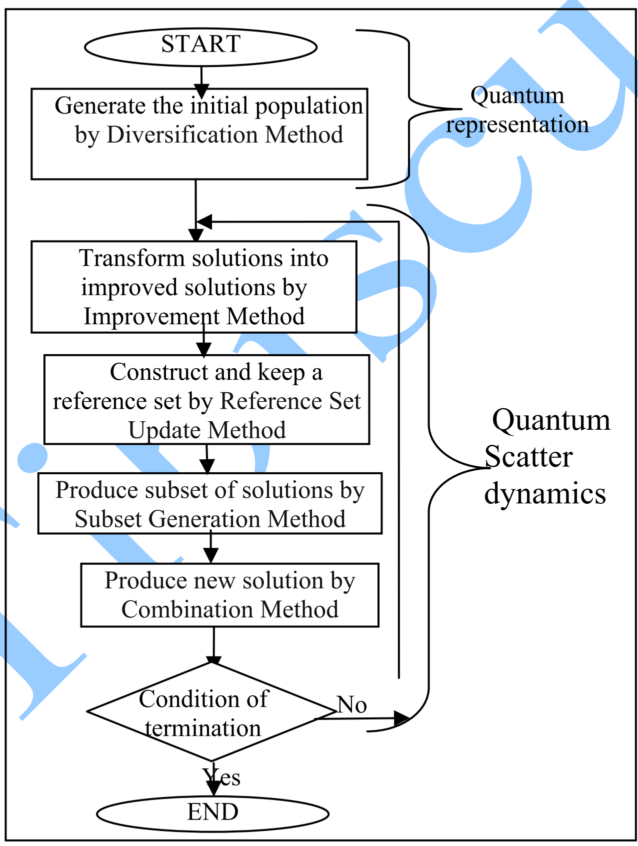

Figure 9. General structure of QSSBDD

5.1 The Diversification Generation Method

The idea behind the diversification generation method is to create a set of diverse solutions. The probabilistic nature of quantum algorithm offers a good way to generate new solutions. Consequently, we apply the measure operation several times to generate the initial set of trial solutions. We can also inject equi-probable quantum solutions (figure 10) in each application of the diversification method. Moreover, we can use a heuristic to construct a good variable order. The BDD heuristics generally uses the information extracted from the Boolean functions such as the semantic significance extracted from Boolean function entries.

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

Figure 10. Equi-probable Quantum solution of BDD variable ordering

5.2 The Improved Method

The Improvement Method (IM) is used to transform the solution generated into an improved solution via a transformation procedure. In our algorithm, we have used several improved operators. The first is the classical quantum mutation seen in the previous section. In addition, we apply another quantum inspired operator which is the quantum interference. This operation is useful to intensify research around the best solution. We have also used another improved method specific to the problem of BDD. Indeed, in order to improve the efficiency of the exploration process, we have introduced the local search method in the QISSA. The basic idea consists in changing the order of two variables by permuting them (figure 11). While applying this technique to several pairs of variables, it helps to choose a variable order which decreases the size of the OBDD. This technique is known under the name of the SWAP method. Moreover, in order to more enhance our approach, we can use the Sitfiting strategy [ISY91]. The sifting algorithm is the best BDD heuristics used to minimize the BDD's size.

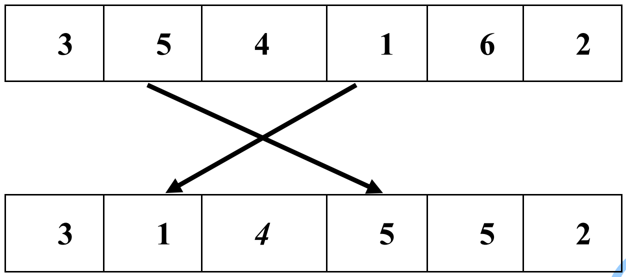

Figure 11. The SWAP method

5.3 The Reference Set Update Method

The aim is to generate a set of both high quality solutions and diverse solutions. The reference set update method follows each application of the improvement method. Generally, it's executed right after the improvement method because of its linking role with the subset generation method. The reference set update method is based on assessing the quality of a candidate solution. In order to avoid a premature convergence of our algorithm, the reference set must preserve both quality and variety. Therefore, the reference set RefSet is divided into two groups: RefSet1 which holds the set of best solutions and RefSet2 which contains the set of bad solutions compared to those of the RefSet1. This step replaces the worst solution with the new best solution found. Consequently, Solutions are included in the reference set by quality or diversity. The assessment of the quality of each solution is done by using the size of the BDD obtained from the variable order as criterion of selection.

5.4 The Subset Generation Method

This method generates new types of solution subsets which are used in the combination method. Each generated subset contains two, three or more solutions. So, we take randomly solutions from the two groups ReftSet1 and ReftSet2 in order to generate new diverse subsets. Therefore, the subset contains the good solutions and the bad solutions.

5.5 The Combination Method

This operator performs combination of solutions. It consists in creation of new solutions by using a linear combination of quantum matrices (fig.12). In our algorithm, we have used simple linear combination operation. We select randomly three quantum matrices. Then, we generate a new quantum matrix from the three matrices. In order to keep the unity characteristic of the qubit, we use the angles represented by each qubit. In fact, we can give to each qubit $Q(a_i,b_i)$ a geometric representation: $Q1(cos(\Phi1),sins(\Phi1))$.

0.5644	0.9219	1.0000	0.2442	0.6121
0.8255	0.3875	0	0.9697	0.7908

Figure 12. An example of linear combination of three quantum matrices

6 Implementation and Evaluation

QSSBDD is implemented in java 1.5 and is tested on a microcomputer with a processor of 3 GHZ and 1 GB of memory. We have used the package JBDD [Wha**] which contains a set of tools for the creation and the manipulation of BDDs. To assess the efficiency and accuracy of our approach several experiments were designed. The experiments were undertaken on a set of tests created randomly with the logic gates AND, XOR and NOT. These tests are selected of such kind to show the performances of our program.

We have compared the performance of the QSSBDD against three evolutionary algorithms based on different local search strategies: QGABDD based on quantum genetic algorithm with simple mutation [LS07], QEASIFT based on quantum evolutionary algorithm with sifting technique and QEASWAP based on quantum evolutionary algorithm with swap technique [LS08]. Moreover, we have compared our results with the state of art WIN3ITE which is based on an iterated version of the window permutation algorithm with windows size k=3. This program is integrated in the popular BDD package Buddy [Rud93]. Finally, Friedman and Wilcoxon matched-pair signed-rank tests were carried out to test the significance of the difference in the accuracy of each method.

The results are summarized in Table 2. The first column shows the benchmark name, the second column show the number of variables in each test, and the others column show the results found by each method in this experiment. The figure 16 shows the Freidman test, the best program is the nearest to 0 (minimization). According to the Freidman test, there is no significant difference between the performances of the methods QSSBDD, QEASIFT and QEASWAP. As we can see, the QEASIFT is the most successful program in this experiment (Figure 13). Unfortunately, it was found that QEASIFT is very slow with the procedure sifting compared to the other methods used in this experiment. On the other hand, the program QSSBDD is better than QEASWAP. Finally, the iterated window permutation method WIN3ITE and the method QGABDD are not successful in this experiment. In most cases, WIN3ITE and QGABDD run worse than the other methods.

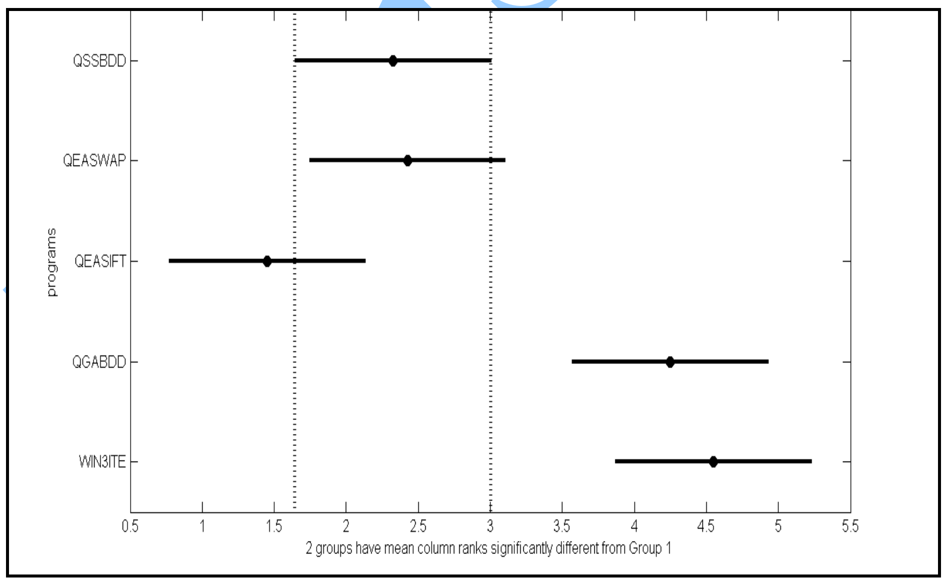

Figure 13. Test of Friedman (a=0.05): comparison between QSSBDD and other programs

Table 2. The results

Test	Number variables	QSSBDD	QEASWAP	QEASIFT	QGABDD	WIN3ITE
Test40_1	40	196	172	108	242	138
Test40_2	40	400	335	268	437	319
Test60_1	60	1127	874	437	1339	2216
Test60_2	60	811	781	883	2402	1451
Test80_1	80	453	453	320	720	1720
Test80_2	80	785	713	579	1499	1990
Test100_1	100	3484	2540	1733	8275	10777
Test100_2	100	2145	1973	1407	4315	3491
Test100_3	100	475	435	399	798	1061
Test100_4	100	3344	2451	2325	8574	8112
Test150_1	150	3128	3128	3385	15984	18330
Test150_2	150	5335	6394	4447	8699	11980
Test150_3	150	1848	1764	1290	3516	4414
Test200_1	200	3963	6059	4021	20273	21966
Test200_2	200	4620	4620	4574	11295	13310
Test200_3	200	3990	5348	4857	27730	36593
Test200_4	200	90299	12478	109760	209380	256879
Test250_1	250	213655	29162	259763	443280	483304
Test250_2	250	13481	13727	9107	21803	25043
Test250_3	250	41743	51535	48359	60451	61518

Conclusion

In this work, we have presented a new evolutionary algorithm based on hybridization between scatter search algorithm and quantum computing called Quantum Inspired Scatter Search Algorithm (QISSA). The quantum representation of the solutions allows the coding of all the potential variable orders with a certain probability. The optimization process consists of the application of a scatter search dynamics enhanced by quantum operations such as the interference, the quantum mutation and measurement. We have explained the basis of QISSA through the resolution of the BDD variable ordering problem. The experimental studies prove the feasibility and the effectiveness of our approach. The proposed algorithm reduces efficiently the population size and the number of iterations to have the optimal solution. Thanks to superposition, interference, and quantum mutation operators, better balance between intensification and diversification of the search is achieved. However, there are several issues to improve our program. Firstly, to raise the speed of our program, it's better to use parallels machines because it was verified effectively that quantum inspired algorithms can work better on parallels machines. Secondly, the performance of the algorithm may be improved by using a clever startup solution.

Reference

[Bry92] **R. E. Bryant** - *Symbolic Boolean Manipulation with Ordered Binary Decision Diagrams*, ACM Computing Surveys, Vol. 24, No. 3, pp. 293-318,1992.

[BW96] **B. Bollig, I. Wegener** - *Improving the variable ordering of OBDDs is NPcomplete,* IEEE Trans. on Comp., 45(9):993–1002, 1996.

[CDM00] **U. S. Costa, D. Déharbe, A. M. Moreira** - *Variable ordering of bdds with parallel genetic algorithms*. In Proceedings of PDPTA'2000, 2000.

[CGL94] **E. M. Clarke, O. Grumberg, D. E. Long** - *Model checking and abstraction,* ACM Transactions on Programming Languages and Systems, 16(5) :1512–1542, September 1994.

[DB98] **R. Drechsler, B. Becker** - *Binary Decision Diagrams: Theory and Implementation,* Kluwer Academic Publisher, 1998.

[FFK88] **M. Fujita, H. Fujisawa, N. Kawato** - *Evaluation and improvements of Boolean comparison method based on binary decision diagrams*. In Int'l Conf. on CAD, pages 2–5, 1988.

[FOH93] **H. Fujii, G. Ootomo, C. Hori** - *Interleaving based variable ordering methods for ordered binary decision diagrams*. In Int'l Conf. on CAD, pages 38–41, 1993.

[GLW00] **F. Glover, A. Løkketangen, D. L. Woodruff** - *Scatter Search to Generate Diverse MIP Solutions*. In: OR Computing Tools for Modeling, Optimization and Simulation: Interfaces in Computer Science and Operations Research, Eds. M. Laguna and J.L. González-Velarde, pp. 299 – 317,2000.

[HK04] **K.H. Han, J.H. Kim** - *Quantum-inspired Evolutionary Algorithms with a New Termination Criterion, Hε Gate, and Two Phase Scheme*, IEEE Transactions on Evolutionary Computation, IEEE Press, vol. 8, no. 2, pp. 156-169, April 2004.

[ISY91] **N. Ishiura, H. Sawada, S. Yajima** - *Minimization of binary decision diagrams based on exchange of variables*. In Int'l Conf. on CAD, pages 472–475, 1991.

[Jae06] **G. Jaeger** - *Quantum Information: An Overview*. Berlin: Springer. 2006

[LS07] **A. Layeb, D. E. Saidouni** - *Quantum Genetic Algorithm for Binary Decision Diagram Ordering Problem*. The proceeding of International Journal of Computer Science and Network Security 7(9), 130–135, 2007.

[LS08] **A. Layeb, D. E. Saidouni** - *Quantum Differential Evolution Algorithm for Variable Ordering Problem of Binary Decision Diagram*. In the Proceedings of the 13th International CSI Computer Conference CSICC'08, Springer: Communications in Computer and Information Science, CCIS Vol. 6, pp. 942–945, Iran Mars 2008.

[LST08] **A. Layeb, D. E. Saidouni , E.-G. Talbi -** *A New Quantum Scatter Search Algorithm for MAX 3-SAT Problem. In the proc. of international workshop of Complexity and Intelligence of the Artificial and Natural Complex Systems*. CANS08, Petru Maior University Press, ISSN 2065-0426, pp.115-120. Romania November 2008.

[MLG06] **R. Marti, M. Laguna, F. Glover -** *Principles of scatter search*, Eur J Oper Res 169 (2006), pp. 359–372.

[Rud93] **R. Rudell** - *Dynamic variable ordering for ordered binary decision diagrams*, In Int'l Conf. on CAD, pages 42–47, 1993.

[TDB04] **H. Talbi, A. Draa, M. Batouche** - *A Quantum-Inspired Differential Evolution Algorithm for Rigid Image Registration*. International Conference on Computational Intelligence. pp. 408-411,2004.

[Wha**] **J. Whaley** - *JAVABDD, a Java Binary Decision Diagram Library*, Stanford University http://javabdd.sourceforge.net