A Quantum Particle Swarm Optimization Approach for Feature Selection in the Data Classification

Sihem Benkhaled
Dept. of computer science
Univ Abbes Laghrour
Khenchela, Algeria
sibenkhaled@gmail.com

Hichem Houassi
Dept. of computer science
Univ Abbes Laghrour
Khenchela, Algeria
houassi_h@yahoo.fr

Mounir Hemam
Dept. of computer science
Univ Abbes Laghrour
Khenchela, Algeria
mounir.hemam@gmail.com

Abstract

Feature selection is a preprocessing step that plays an important role in Data mining. It allows searching a reduced size of features' subset from a large set, by eliminating redundant and irrelevant features. These are often used to perform the supervised classification task, in order to maintain or improve classifier performance. The search for a features' subset is an NP-difficult optimization problem, which can be solved by metaheuristics; we are interested by the metaheuristics based on swarm intelligence for feature selection problem. In this paper, we propose a "Binary Clonal Quantum Particle Swarm Optimization" algorithm, denoted BC-QPSO for selecting a subset of relevant features. This algorithm is developed from hybridization between an "optimization of the binary quantum particle swarm (BQPSO)" metaheuristic and an artificial immune system using its clonal selection algorithm (CSA). The experiments are carried out using UCI databases (University of California, Irvine). Besides, the experimental results of our approach are compared with those obtained by two approaches proposed in the literature: "Binary Particle Swarm Optimization (BPSO)" and "Binary Quantum Particle Swarm Optimization (BQPSO)", the results obtained are competitive and show the efficiency of our algorithm BC-QPSO.

Keywords—Feature selection- Classification- Metaheuristics- Particle Swarm Optimization (PSO).

1. Introduction

Data classification is the most important task of Data mining; it consists in studying the characteristics of a new object to give it a predefined class. It is characterized by a definition of specific classes and a set of examples previously classified, the goal is to create a model that can be used to classify unclassified data [Dan05].

In Data mining, feature selection is a process that searches for a subset of relevant features in datasets, in order to improve classification performance [Das97]; the notion of relevant subset of features depends on the objectives and criteria of the constructed classification model [Blu97].

The search for an optimal subset of features is considered as NP-difficult optimization problem [Blu97], in the literature, different methods have been developed to solve this problem; we can classify them into classical and metaheuristic methods [Cot03].

In this paper we will focus on metaheuristics methods that follow a powerful mechanism to achieve better solutions, rather than heuristics, which just propose solutions to a problem without guaranteeing that they will be best ones [Zhao13]. There are many optimization metaheuristic algorithms employed in feature selection, such as genetic algorithm (GA) [Nun02], clonal selection algorithm (CSA) [Nun02], ant colony optimization (ACO) [Dor04] and particle swarm optimization (PSO) [Ken95]. PSO is a newest metaheuristic, despite that it proved its simplicity and efficiency compared to other methods [Zhao13, it is important to try to improve this method.

The proposed approach (BC-QPSO) combines the quantum binary variant of the PSO algorithm with the clonal selection algorithm, which allows a good exploitation of the space research [Nun02].

To assess BC-QPSO performance and approve its efficiency, we compare it with two algorithms proposed in the literature "Binary Particle Swarm Optimization (BPSO)" and "Quantum Binary Particle Swarm Optimization (QBPSO)" [Zhao13]. This approach aims on the one hand to reduce the size of the features' set used and on the other hand to improve the classification task (increase classification rate), with a rapid convergence towards the global optimum.
This paper is organized as follows: The next section explains PSO and Clonal selection algorithm. Section 3 presents the approaches presented in the literature related to the problem of hybridization between PSO and other approaches. In section 4, our BC-QPSO approach is presented. Results and discussion are shown in section 5. Finally the conclusion and future works are presented in section 6.

2. Prerequisites

In this section we give an overview of the concepts related to the understanding of the proposed approach, it is about the particle swarm optimization (PSO) method and one of its extension (QBPSO), as well as the Clonal Selection Algorithm (CSA), for further details, reader may consult [Ken95], [Zhao13] and [Nun02].

2.1. Quantum Binary Particle Swarm Optimization (QBPSO)

The particle swarm optimization algorithm has been proposed by Kennedy and Eberhart in 1995, developed as a stochastic and iterative algorithm, inspired by a social behavior of animals evolving into swarms [Ken95]. This algorithm involves a set of agents for solving a given problem; this set is called a swarm, the swarm is composed of a set of members called particles [Ken95].
The swarm of particles flies over the space research, to get the global optimum (best particle's position), the particles' position represent potential solutions to the problem being treated. The displacement of each particle is influenced by its velocity, the best position that has been retained (Pbest) and the best position known by all the particles of the swarm (Gbest) [Ken95].
This algorithm has the disadvantage of not giving rise to sufficient exploration, which can lead to stagnation in a local optimum and thus premature convergence [Lia06]. In fact, several extensions have been proposed in the literature, we are interested in this work by: **Quantum Binary Particle Swarm Optimization (QBPSO).**
It is a variant of PSO, suitable for solving discrete or binary optimization problems, including the principles of quantum mechanics, in the classical model of the PSO, the particle moves taking into account its current position in the space research and its velocity of displacement, which is not the case in the QBPSO model where the velocity does not influence on the displacement of the particle, the particles in this case are represented by a binary vector which represents the position of each one [Zhao13].
In fact, in the quantum binary version of the PSO algorithm the particle moves and changes its position according to the following equations [Zhao13]:

- The position of the particle is represented as a binary chain: $X_i = (X_1, ...,X_D)$
  The distance between two chains (positions) is calculated by a distance of Hamming, which consists in calculating the number of times, where the two chains are different bit by bit, calculated by:

$$|X - Y| = d_H(X, Y)$$
$$\text{Or : } d_H(X, Y) = b \quad \text{with :}$$
$$b = d_H(X_i, P_i) = ß(d_H(X_i, m_{best}))\ln(1/u) \qquad (1)$$

ß, u: Random numbers;
m_{best} : calculated by Procedure 2

- We use also the two following procedures:

Pi obtained by a crossing operation between P_{besti} and G_{best}, calculated by the procedure:

Get_P(P_{besti}, G_{best})
Apply a crossover operation between P_{besti}, G_{best} to generate two binary vector z1et z2 ;
If rand( )<0.5 **Then**
Pi = z_1 ;
Else $P_i = z_2$;
EndIF
Return P_i ;

Procedure 1

m_{best} calculated by the procedure: Get_m_{best}(P_{best})

For j=1 to d (d : taille de P_{best}) **Do**
sum=0 ;
For all particle i **Do** sum=sum+P_{best}[i][j] ; **EndFor**
avg=sum/M ;
If avg >0.5 **Then** m_{best}[j]=1 ;**Finsi**
If avg > 0.5 **Then** m_{best}[j]=0 ;**Finsi**
If avg=0.5 **Alors If** rand( )> 0.5 **Then** m_{best}[j]=0 ; **Else** m_{best}[j]=1 ;
EndIf EndIf
EndFor
Return m_{best} ;

Procedure 2

- Calculate P_{best} and G_{best} (best position recognized by each particle and for all particles in swarm, respectively):

$$P_{besti}(t+1) = \begin{cases} x_i(t+1) & \text{if } f(x_i(t+1)) >= f(P_{besti}(t)) \\ P_{besti}(t) & \text{else} \end{cases} \quad (2)$$

$$G_{best}(t+1) = argmax_{Pbesti} f(P_{besti}(t+1)) \quad , 1 <= i <= N \quad (3)$$

The update of Xi obtained by a mutation of P_i in procedure 3 (**Transf**) with the probability Pr:

$$Pr = \begin{cases} 1 & \text{if } b/l > 1 \\ b/l & \text{else} \end{cases} \quad , l : \text{size of position chain} \quad (4)$$

Transf(P_i,**Pr**)

For all bit of P_i **Do**
  If rand()<Pr **Then**
    If bit stat is 1 **Then** reset to 0 ; **If** reset to 1 ; **EndI; EndIf;**
EndFor
$X_i=P_i$;
Return X_i;

Procedure 3

We note that the **Objective function (fitness function) - f(x_i)**, is a function that serves as a criterion for determining the best solution to an optimization problem. In concrete terms, it evaluates problem's solutions by associating to each solution a value. Each problem can define its own function, the goal of the optimization problem is then to minimize or maximize this function (value of function) up to the optimum [Ken95].

The general algorithm runs as follows [Zhao13]:

Algorithm 1: Quantum binary particle swarm optimization algorithm

1. Initialize randomly the position x_i of N particles i;
2. Calculate their affinities (Evaluate the positions with objective function $f(x_i)$);
3. **While** stop condition is not verified **Do**
   a. Change particles' position by (**Transf** procedure);
   b. Calculate $f(x_i)$ for each particle i (Evaluate the particle's position);
   c. Update P_{besti} for each particle i by (2);
   d. Update Gbest by (3);
   End while
4. Return G_{best};

2.2. Clonal selection algorithm (CSA)

Clonal selection used by the natural immune system to define the basic features of an immune response to an antigenic stimulus. It establishes the idea that only those cells that recognize the antigens are selected to proliferate; the selected cells are subject to an affinity maturation process, which improves their affinity to the selective antigens [Nun02].
Clonal selection algorithm is an abstraction of the mechanisms of memory of immune systems, whose problem to be treated plays the role of an antigen [Nun02].
The possible solutions play the role of the antibodies of a B cell called B lymphocyte, when the antibodies of a B cell encounter an antigen (good antibodies), the B cell begins to create clones (copies); the latter undergo mutations to improve their affinities by adapting their antibodies to invading antigens (doing some changes to the solution) [Nun02].
The general algorithm runs as follows [Nun02]:

Algorithm 2: Clonal Selection Algorithm

1. Initialize a population of N antibodies;
2. Calculate their affinities (Evaluation);
3. **While** stop condition is not verified **Do**
   a. Produce cell clones with good affinities (Create copies);
   b. Mute the clones produced (Change some characteristics);
   c. Select cells with best affinities
   d. Replace best affinity cells with the selected cells;
   End while
4. Return the best solutions;

3. Overview of Metaheuristics for Feature Selection based on PSO

Table 1 represents several hybridizations between PSO and other approaches, these methods have been proposed in the context of feature selection, in order to improve the PSO method.

Table 1: Hybridizations between metaheuristics based on PSO.

Approach	Principe	Authors
(HGPSO)	The individuals of a new generation are created by the transactions of growth, mutation and PSO.	C-F.Juang [Jua04]
PSO-GA 1	Use fuzzy logic to integrate the results of both methods.	F.Valdez, P.Melin, O.Castillo [Val09]
PSO-GA 2	The position update of the best particles made by GA.	K.Premalatha, A.M.Natrajan [Pre09]
PSO-GA 3	Replace the last step of AG (mutation) with PSO.	H.Hachimi [Hac13]
PSO-GA 4	PSO with mutation operation.	Zhang [Zha05]
GSO	- Population divides into two subpopulations. - Follow GA and PSO mechanism for each iteration. - Sub-population will be grouped to update the initial population. - Initial population will be divided again and restart the algorithm.	Gandelli, Genetical [Gra06]
NPSO	Integrate the operations (crossing, mutation) in the steps of the PSO stadard version.	Lian [Lia06]
PSO1	Implement the algorithm proposed to generate the construction factor value K from the interval [0.2.0.9] at each iteration.	Achtning [Ach08]
PSACO	ACO for update best position of each particle.	P.S.Shelokar, V. K. Jayaramen [She07]
PSO-ACO 1	ACO to replace the best position of each particle in PSO.	P.S.Shelokar, V. K. Jayaramen [She07]
PSO-ACO 2	ACO apply to perform a local search in which ants are guided by the concentration of pheromones to update positions calculated by PSO.	Habibi [Hab06]
PSO-ACO-simulated annealing	ACO used to replace the position of each particle found by PSO. Simulated annealing to control the exploitation of the best particle of the group.	Habibi [Hab06]
PSO-VSN	PSO with crossover and local search and variable neighborhood.	Czgolla,Fink [Czg08]
PSO-SA	During the PSO algorithm, appeals to SA in mutation and crossover.	X.Shen [She07]
PSO-local search	PSO local search method to improve the search locally as they used the GA concepts to better explore the space research.	Lope, Coelho [Lop05]

In these hybridizations (Table 1), the proposed approaches try to improve the efficient of PSO algorithm; but in some cases PSO can easily fall into the valley of the local optimum during several iterations. This phenomenon is known as stagnation in a local optimum [Val09].
We conclude that there are no hybridizations between the two algorithms (PSO and CSA), for that raison and in order to contribute to improve PSO method, we thought to combine PSO with CSA to explore the space research better by benefiting from clonal algorithm mechanisms [Nun02].

4. Proposed BC-QPSO Approach

According to the proposed approaches in literature, we noted the absence of hybridizations between the BQPSO approach and Clonal Selection Algorithm, in the context of feature selection problem, in order to improve classification performance.
For this reason, we proposed BC-QPSO (Binary Clonal- Quantum Particle Swarm Optimization) as a hybridized approach to better exploit the space research and find best solution. Figure 1 illustrates a description of BC-QPSO feature selection algorithm.

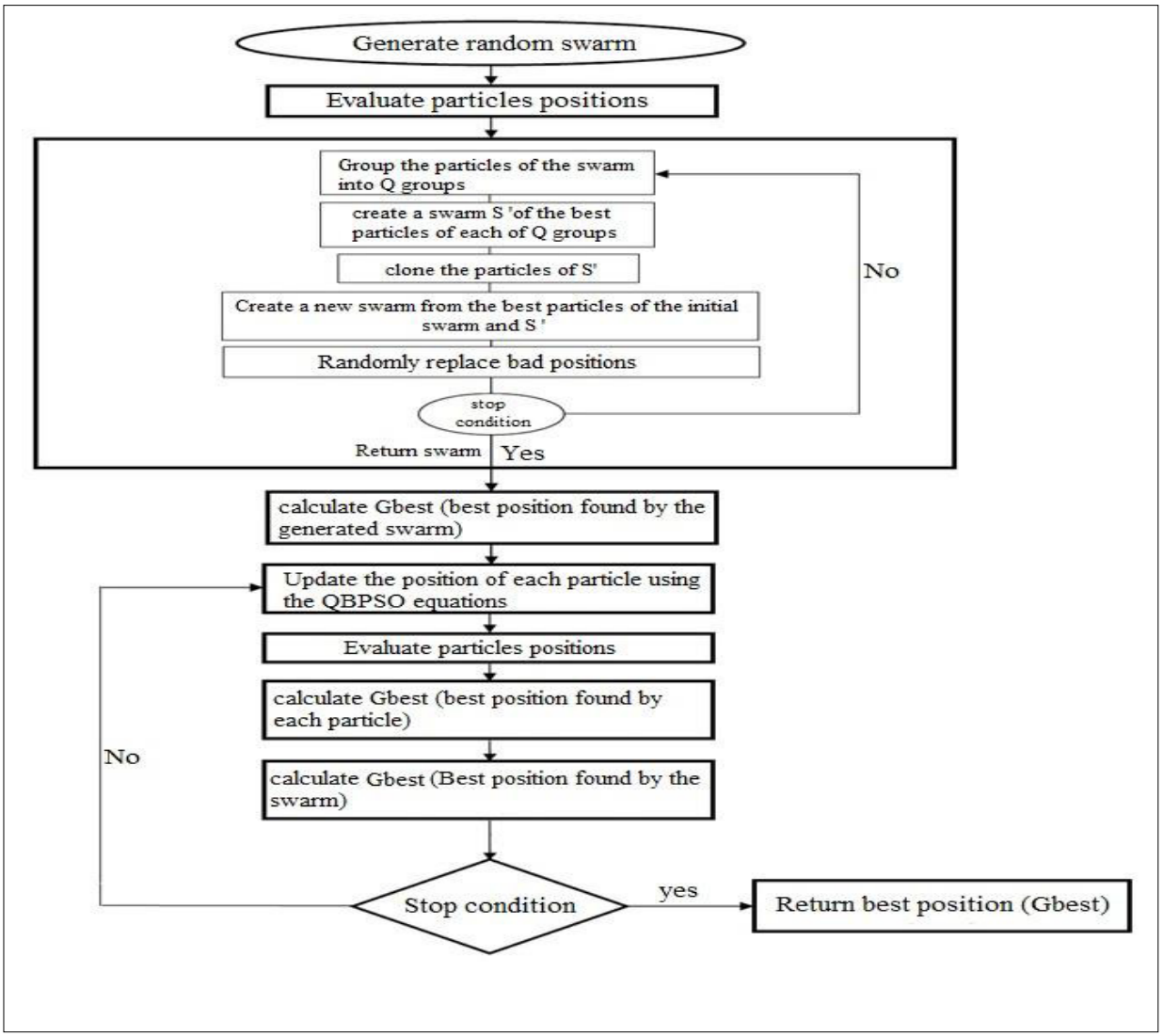

Figure 1: BC-QPSO flowchart.

4.1. BC-QPSO approach steps:

The goal of hybridization is to improve the search performance and the rapid convergence towards the best solution (select a relevant subset of features). In our case, we consider the best solution (relevant features' subset) that is the one obtained by a high classification rate in classification task.
The idea is to benefit from the advantages of the two algorithms, the clone selection algorithm (CSA) and the BQPSO algorithm. Indeed, the objective of our approach is to improve the quality of the swarm by a new strategy initialize the population (particles).

We use the operations of the clone selection algorithm into the QBPSO algorithm, in order to ensure a good coverage of the space research by the particles. In other words, instead of initializing a swarm randomly, an improved swarm is created by the following steps:

Step 1: Generation of the initial population:
The P vectors that represent positions' particles are randomly generated; create an initial S swarm of the space research.
Step 2: Evaluation of particles:
For all the particles of a swarm S, calculate the fitness function value by the accuracy of the classifier (classification rate), using the subset of features associated to each positions. The classifier induced in the experiments is a KNN[1].
Step 3: Improve the quality of the swarm S:
To improve the swarm S, we generate new population of particles according to the algorithm inspired by the mechanisms of the clonal algorithm (see Algorithm 3 below).
Step 4: Update and evaluate the position of each particle following the steps of QBPSO approach, using the equations presented in (section 2.1), repeat this step until stop criterion is verified, see Figure 1 (section 4).

Algorithm 3: Generation of the particles' population

1. Input: S1 initial swarm with the value of the fitness functions of each particle;
2. Output: S2 swarm; //improved swarm
3. While stop condition is not verified **do**
a. Group the S1 particles into P groups based on the proximity criterion (each group represent an interval of the fitness function values);
b. Choose the best particles in each of the P groups and create a sub swarm SQ;
c. Clone the particles of SQ; // (used selective-crossover[2] operation between particles' positions pair to pair)
d. Create the new generation of S2 swarm from the best particles of SQ and S1;
e. Replace the bad particles (low fitness function values) of the new S2 swarm randomly;
f. Replace S1 with S2;
End While

4.2. Data encoding

Each particle has one position, the position is represented by a binary vector, the bits equal 1 to the selected attributes, and the bits equal 0 to the unselected attributes [Sun07].
Example: Let the particle position i represented as follows:

$$Xi = [1, 1, 0, 1, 0, 1]$$

The size of the vector corresponds to the number of dimensions (all attributes in the database).

4.3. Fitness function (Objective function)

To evaluate particles' position, we use the classification rate, this rate found by applying the KNN (Datamining algorithm) to the datasets (using the subset of features selected). The speed execution of KNN makes it the most appropriate to our approach BC-QPSO; it is applied in each iteration. In our study, the classification rate is considered as a function of fitness (objective function f (xi)); the best solution is the one that has a higher classification rate.

5. Experiment results

We implement this approach with Java using Data mining software **Weka**[3] for classification task. For four datasets, we conduct a performance comparison between our approach BC-QPSO and those proposed in literature: BPSO and BQPSO. All three approaches are applied to four databases by setting the following parameters:

- Stop Criterion: as mentioned above, convergence to the overall solution is not guaranteed in all cases, even if the experiments show the great performance of the method [VAL09].

[1] https://mrmint.fr/introduction-k-nearest-neighbors
[2] https://en.wikipedia.org/wiki/Crossover_(genetic_algorithm)
[3] https://www.cs.waikato.ac.nz/ml/weka/

Therefore, it is quite advised to endow the algorithm with an exit point by defining a number of iterations as stop criterion.

- Size of the space research: depends essentially on the number of particles allocated to finding a solution for a problem.
  There are no rules for determining these parameters; many tests allow having the necessary experience to set these parameters, after several trials.
- Comparison criteria are:
  1. Average number of attributes.
  2. Max classification rate.

➢ Using these two comparison criteria, Table 2 gives the best solution (Max classification rate) obtained for each datasets; the following results are obtained by taking the best solution after 20 iterations and using 20 particles as population size.

- In (Table 2) the comparison is based on the reduced number of attributes relative to the set of original attributes, and the maximum classification rate achieved by the learning model with preserving the selected relevant attributes.
- The BC-QPSO approach succeeds in reducing the size of the set of attributes by finding the optimal small subassembly, and increases the classification rate (Global optimum) compared with two other approaches, so it goes out somewhat from the local optimum.
- We also note that BPSO is competitive with BC-QPSO in terms of the size of solution found and the max classification rate.

➢ Figure 2 represents an Evolution of max classification rate relative to the number of iterations, by setting the population size to 20 particles:

- BC-QPSO has a fast convergence towards the global optimum, competitively with other approaches.
- Figure 2 shows that in most cases, we obtain the optimal solution after a 10^{th} or 20^{th} iteration, which proves the effectiveness and the speed of our algorithm.
- In fact, the results obtained on the **Semeon** dataset show that our approach converges to the global optimum in all of runs and indicates that BC-QPSO can be applied for large number of features.

Table 2: Results Comparison of implemented approaches.

Datasets	**Total number of attributes**	**Approach**	**Average number of attributes selected**	**Max classification rate**
Semeon	257	**BPSO**	128.2	92.66
		BQPSO	130.1	92.54
		BC-QPSO	129.6	92.68
MUSK	168	**BPSO**	80.26	99.15
		BQPSO	82.83	98.94
		BC-QPSO	82.56	99.16
LIBRAS	91	**BPSO**	44.16	89.16
		BQPSO	44.03	88.88
		BC-QPSO	42	90.87
Audiology	70	**BPSO**	36.7	77.43
		BQPSO	43.86	77.87
		BC-QPSO	35.73	79.99

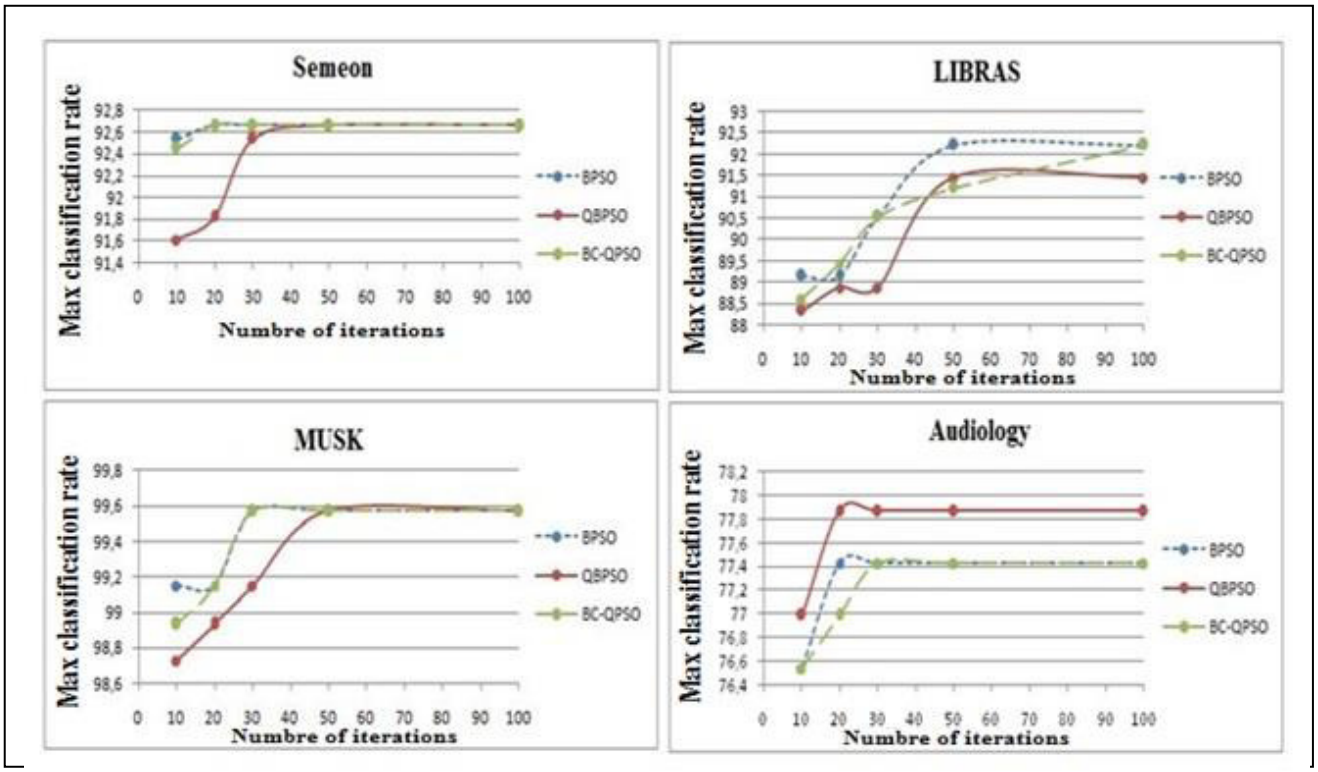

Figure 2: The convergence of BC-QPSO, BQPSO and BPSO according to the number of iterations.

6. Conclusion

This paper proposed a new approach in the context of feature selection by improving a metaheuristic algorithm; which is quantum binary particles swarm optimization (QBPSO). The proposed approach, called BC-QPSO combined BQPSO and the Clonal Selection Algorithm (CSA).

Results of this approach have been compared to other approaches, such as QBPSO and BPSO, the principal comparison criterion was the classification rate. We have applied the algorithms to some of datasets randomly selected. The results have shown the efficiency of BC-QPSO and its competitiveness, compared to the two others approaches, in terms of classification rate and in the reduction of the size of the features' subset. BC-QPSO also solved somewhat the stagnation in a local optimum problem, by improving the classification performance (increase greatly the classification rate comparing with BPSO and BQPSO), but not totally.

In future works, we plan to improve our approach by using the cooperative approach in "Cooperative Binary Quantum Particle Swarm Optimization (CBQPSO)" presented in [Zhao13]. This will update Pbest and Gbest in each step of our BC-QPSO algorithm.

References

[Ken95] J. Kennedy and R.C. Eberhart, Particle Swarm Optimization, In: Conference proceeding of neural networks, Perth. Australia : IEEE, p. 1942–1948, nov. 1995.

[Blu97] A.L.Blum and P.Langley, Selection of relevant features and examples in machine learning, In: Artificial Intelligence, p. 584-594, 1997.

[Das97] M.Dash and H.Liu., Feature selection for classification, In: Intelligent Data analysis 1, p. 131–156, 1997.

[Cle02] YM.Clerc and J.Kennedy, The particle swarm : explosion, stability, and convergence in multidimensional complex space , In : IEEE Transactions on Evolutionary Computation 6, p. 58–73, 2002.

[Nun02] L.Nunes de Castro and F.J.Von Zuban., Learning and optimization using the clonal selection principle, In: IEEE Transactions on Evolutionary Comptation 6, p. 239–251, 2002.

[Cot03] C.Cotta and P.Moscato, The k-features set problem is W[2]-complete, In: Journal of computer and system sciences, p.686-690, 2003.

[Dor04] M.Dorigo and T.Stützle. Ant Colony Optimization, Cambridge: MIT Press, 2004.
[Jua04] C.F.Juang, A hybrid of genetic algorithm and particle swarm optimization for recurrent network design, In : IEEE Trans Syst Man Cybern 34, p. 997–1006, 2004 .
[Dan05] Daniel T.Larose, Discovery knowledge in data, Canada, 2005.
[Lop05] HS.Lopes and LS.Coelho, Particle swarm optimization with fast local search for the blind travelling salesman problem, In : Proceedings of fifth international conference on hybrid intelligent systems, p. 6–9, 2005.
[Zha05] Y.N.Zhang, Q.N.Hu and H.F.Teng., Active target particle swarm optimization, In : Journal of concurrency computation practices and experiences 20, p. 29–40, 2005.
[Hab06] J.Habibi, SA.Zonouz and M.Saneei, A hybrid PS-based optimization algorithm for solving traveling salesman problem, In : IEEE symposium on frontiers in networking with applications, p. 18–20, 2006.
[Gan06] Gandelli and al, Genetical swarm optimization : an evolutionary algorithm for antenna design, In : Journal of Automatika 47, p. 3–4, 105–112, 2006.
[Lia06] Z.Lian, X.Gu and B.Jiao, A similar particle swarm optimization algorithm for permutation flow shop scheduling to minimize makespan , In: Applied Mathematics and Computation , p. 773–785, 2006.
[Sun07] J.Sun and al, Quantum-Behaved Particle Swarm Optimization with Binary Encoding , In : Center of Intelligent and High Performance Computing, p. 376–385, 2007.
[She07] P.S.Shelokar and V.K.Jayaramen, Particle swarm and ant colony algorithms hybridized for improved continuous optimization, In : Applied Mathematics and Computation 188, p. 129–142, 2007.
[Ach08] J.Achtnig, Particle Swarm Optimization with Mutation for High Dimensional Problems, In : Engineering Evolutionary Intelligent Systems-Springer, p. 423–439, 2008.
[Czo08] J.Czogalla and A.Fink, On the Effectiveness of Particle Swarm Optimization and Variable Neighborhood Descent for the Continuous Flow-Shop Scheduling Problem, In: Metaheuristics for Scheduling in Industrial and Manufacturing Applications, p. 61–89, 2008.
[Pre09] K.Premalatha and A.M.Natarajan, PSO with GA Operators for Document Clustering, In : International Journal of Recent Trends in Engineering 16, 2009.
[Val09]F.Valdez, P.Melin et O.Castillo, A New Evolutionary Method Combining Particle Swarm Optimization and Genetic Algorithms Using Fuzzy Logic, In : Soft Computing for Hybrid Intelligent Systems, p. 347–36, 2009.
[Hac13] H.Hachimi, Hybridations of algorithms metaheuristics in global optimization and its applications, thesis Ph.D, Mohamedia ingéniering school , (Rebat,Maroc), 2013.
[Zhao13] J.Zhao, J.Sun and W.Xu, A binary quantum-behaved particle swarm optimization algorithm with cooperative approach, IJCSI, Vol. 10, Issue 1, No 2, January 2013.
[Val14] F.Valdez, P.Melin and O.Castillo, A New Evolutionary Method Combining Particle Swarm Optimization and Genetic Algorithms Using Fuzzy Logic, In : Soft Computing for Hybrid Intelligent Systems, p. 347–361, 2014.