2009 International Nuclear Atlantic Conference - INAC 2009
Rio de Janeiro,RJ, Brazil, September27 to October 2, 2009
ASSOCIAÇÃO BRASILEIRA DE ENERGIA NUCLEAR - ABEN
ISBN: 978-85-99141-03-8

AN APPROACH USING QUANTUM ANT COLONY OPTIMIZATION APPLIED TO THE PROBLEM OF IDENTIFICATION OF NUCLEAR POWER PLANT TRANSIENTS

Márcio H. Silva, Roberto Schirru and Jose A.C.C. Medeiros.

Laboratório de Monitoramento de Processos – PEN/COPPE
Universidade Federal do Rio de Janeiro - UFRJ
Ilha do Fundão, s/n
21945-970 Rio de Janeiro, RJ

marciohenrique@lmp.ufrj.br
schirru@lmp.ufrj.br
canedo@lmp.ufrj.br

ABSTRACT

Using concepts and principles of the quantum computation, as the quantum bit and superposition of states, coupled with the biological metaphor of a colony of ants, used in the Ant Colony Optimization algorithm (ACO), Wang et al developed the Quantum Ant Colony Optimization (QACO). In this paper we present a modification of the algorithm proposed by Wang et al. While the original QACO was used just for simple benchmarks functions with, at the most, two dimensions, QACO_Alfa was developed for application where the original QACO, due to its tendency to converge prematurely, doesn't obtain good results, as in complex multidimensional functions. Furthermore, to evaluate its behavior, both algorithms are applied to the real problem of identification of accidents in PWR nuclear power plants.

1. INTRODUCTION

The evolutionary algorithms [1] are search and optimization tools which act on a population of possible solutions. They are based on the biological metaphor of Darwinian principle of survival of the fittest to produce successively better approximations to a solution over the generations in which the population evolves. At each generation a new set of approximations is created, using the individual's choice criterion by their level of fitness.

The use of some of the concepts of quantum computing [2] associated with the structure of evolutionary algorithms gives rise to Quantum inspired Evolutionary Algorithms (QEA) [3]. It uses a probabilistic representation, known as quantum bit (or q-bit), as its fundamental unit of information. An individual is represented by a string of q-bits allowing the QEA to benefit from the linear superposition of states concept, according to which a single individual is capable to represent several possible solutions, in a binary search space according to the number of q-bits that compose the string.

The Ant Colony Optimization algorithm (ACO) [4] is based on the biological metaphor of the collective learning of a colony of ants that, according to the literature [5], can find the shortest path for a source of food through the reinforcement of a substance known as pheromone. The concept of collective learning is based on the social aspects of intelligence, which means the ability of individual to learn with their own experience in a group [6]. In the ACO it is the pheromone that represents the learning of the artificial ants.

Inspired on the QEA, Wang et al [7] introduced the representation of the q-bit in the ACO to create a new evolutionary optimization algorithm that was denominated Quantum Ant Colony Optimization, the original QACO. In QACO the pheromone is represented by a string of q-bits that allow it to solve continuous problems. However, aren't mentioned the use of a heuristic or evaporation method and, according to the tests accomplished in this work, the algorithm presents a tendency to converge prematurely for multidimensional complex problems, where the number of bits is high or when the number of ants is larger than proposed by the authors.

This work also presents a modified version of the original QACO that uses a new form of updating the q-bits besides a pseudo-evaporation phase in the attempt to overcome the limitations of the original QACO. This new version, that we denominated QACO_Alfa, was applied to some benchmarks and to a real problem of identification of accidents in PWR nuclear power plants with the aim of comparison of its results with the original version of QACO.

The method used for the resolution of the problem of identification of accidents in a PWR nuclear power plant is until then based on the comparison of the Euclidian distance among the curve of an event unknown to the curves of known design basis accidents [8]. An event will be classified as being an accident of the type to which it has the smaller distance.

The paper is organized as follows: section 2 presents a theoretical summary on the evolutionary algorithms involved in the work: the ACO, QEA and original QACO; section 3 describes the modified algorithm QACO_Alfa; section 4 describes the benchmarks functions and the problem of identification of accidents; section 5 displays the results of the tests performed for comparison among original QACO, QACO_Alfa and GA in the case of the problem of identification of accidents; finally section 6 presents the conclusions and final considerations.

2. THE EVOLUTIONARY ALGORITHMS

2.1. Ant Colony Optimization

The Ant Colony Optimization algorithm (ACO) [4] was developed to solve complex combinatorial problems as the Traveling Salesman Problem (TSP) [9]. TSP is the problem to find the smallest route passing by all the cities once and returning to the departure city at the end of the itinerary. Formally, being G a complete graph $G = (V, A)$, where V is a complete set of n vertexes and A is the set of edges that links each pair of cities $i, j \in V$, its formulation can be presented as a minimization problem:

$$Min \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} X_{ij} \tag{1}$$

Subject to restrictions:

$$X_{ij} \in \{0,1\} \qquad i, j = 1,..., n \tag{2}$$

$$\sum_{i=1}^{n} X_{ij} = 1 \qquad i = 1,..., n \tag{3}$$

$$\sum_{j=1}^{n} X_{ij} = 1 \qquad j = 1,..., n \tag{4}$$

$$\sum \sum X_{ij} \leq |S| - 1 \qquad \forall S \subset V, S \neq \emptyset \tag{5}$$

The integer variable $X_{ij} = 1$ indicates that city j is visited immediately after city i. Otherwise, $X_{ij} = 0$. V is the set $\{1, 2,..., n\}$ of cities of the problem and S it is one of their subsets. $|S|$ it represents the cardinality of S. Restrictions (2) and (3) guarantee that for each city i there exists just one arrival connection and one exit connection. Condition (5) guarantees that it does not exist sub-routes, in other words, routes that do not include all the n cities.

In the ACO, artificial ants cooperate to solve the problem evolving along the generations through the collective learning of the colony. To solve the TSP, the artificial ants need to choose the cities that still were not visited using a transition of states rule [10] given by equations (6) and (7) as follows:

$$S = \begin{cases} \max\{[FE(r,s)]^{\delta}[HE(r,s)]^{\beta}\} & \text{if } q \leq q_0 \\ Pk & \text{if } q > q_0 \end{cases} \tag{6}$$

where *FE(r,s)* is a positive real value that represents the amount of pheromone associated to the edge (*r*,*s*); *HE(r,s)* is the value of the relative heuristic function when moving from city r for the city s; parameters δ and β weigh, respectively, the relative importance of the ants learning *FE(r,s)* and of the heuristic knowledge given by the heuristic function *HE(r,s)*; q is a random value with uniform probability in the interval [0,1] and q_0 ($0 \leq q_0 \leq 1$) is a parameter of the algorithm. *Pk* is a variable chosen at random, as function of q, which gives the probability of an ant in city r chooses to go for the city s, according to equation (7)

$$Pk = \begin{cases} \dfrac{[FE(r,s)]^{\delta}[HE(r,s)]^{\beta}}{\sum_{s \in J_k(r)} [FE(r,s)]^{\delta}[HE(r,s)]^{\beta}} & se \quad s \in J_k(r) \\ 0 & otherwise \end{cases} \tag{7}$$

where *FE(r,s)* and *HE(r,s)* as well as δ and β, have the same meaning of that in equation (6), and $J_k(r)$ is the list of cities not already visited by an ant that is in the city r.

Cooperation among ants is expressed by modification of the pheromone values *FE(r,s)* seeking to favor the discovery of good solutions. That modification updates the amount of pheromone in two ways, according the following updating rule equations:
Local updating rule:

$$FE(r,s) = (1-\rho)FE(r,s) + \rho FE_{zero} \tag{8}$$

where ρ is the parameter of pheromone evaporation and FE_{ZERO} is the initial amount of pheromone. The local updating rule is applied after all the ants have used the rule of state transition and after all of them have chosen the next city to be visited. The local updating rule is applied while the solution is being built.

Global updating rule:

$$FE(r,s) = (1-\alpha)FE(r,s) + \alpha(w/bfit) \tag{9}$$

where α is the pheromone evaporation parameter, w is a user defined parameter that associated to parameter α expresses the average learning of the algorithm, and *bfit* is the best fitness of the current configuration. The global updating rule is applied after all the ants have built a complete route and, for this reason, it is considered the learning reinforcement of the algorithm

2.2. Quantum Evolutionary and Quantum Ant Colony Optimization Algorithms

The QEA algorithm [3] uses a new form to represent the binary state, the q-bit

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} \tag{10}$$

where α e β are complex numbers, $|\alpha|^2$ is the probability that the q-bit is found in the state "0" and $|\beta|^2$ is the probability that the q-bit is found in the state "1", obeying the restriction

$$|\alpha_i|^2 + |\beta_i|^2 = 1 \qquad i = 1, 2, ..., m \tag{11}$$

In QEA, an individual is represented by a string of q-bits that is capable to represent 2^m states

$$r_i = \left[\begin{array}{c|c|c|c} \alpha_1 & \alpha_2 & \cdots & \alpha_m \\ \beta_1 & \beta_2 & \cdots & \beta_m \end{array}\right] \tag{12}$$

For exemplo, the individual Q-bit with $m = 3$ below,

$$\left[\begin{array}{c|c|c} \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_1 & \beta_2 & \beta_3 \end{array}\right] = \left[\begin{array}{c|c|c} \frac{1}{3} & \frac{\sqrt{2}}{2} & \frac{-\sqrt{3}}{2} \\ \frac{2\sqrt{2}}{3} & \frac{-\sqrt{2}}{2} & \frac{1}{2} \end{array}\right] \tag{13}$$

is capable to represent the 2^3 states described as follows

$$-\frac{\sqrt{6}}{12}|000\rangle + \frac{\sqrt{2}}{12}|001\rangle + \frac{\sqrt{6}}{12}|010\rangle - \frac{\sqrt{2}}{12}|011\rangle - \frac{\sqrt{3}}{3}|100\rangle + \frac{1}{3}|101\rangle + \frac{\sqrt{3}}{3}|110\rangle - \frac{1}{3}|111\rangle \tag{14}$$

This means that the probabilities of being in the states $|000\rangle$, $|001\rangle$, $|010\rangle$, $|011\rangle$, $|100\rangle$, $|101\rangle$, $|110\rangle$ e $|111\rangle$ are 1/24, 1/72, 1/24, 1/72, 1/3, 1/9, 1/3, 1/9, respectively.

Based on QEA, the q-bit representation was introduced in the ACO to create the Quantum Ant Colony Optimization algorithm (QACO) [10]. The quantum ants work in optimization problems in the binary space allowing them to solve continuous real space problems, different from the ACO, that stands out in the resolution of combinatorial complex problems but is not indicated for the real space problems.

In order that the QEA or QACO algorithms be capable to solve the optimization problems it is necessary to convert the notation of q-bits to the conventional binary form. This is done observing the q-bit, in other words, a binary individual *r* is generated starting from the individual Q-bit through the following equation:

$$r_i = \begin{cases} 0 & if\ |\alpha_i|^2 > \eta \\ 1 & if\ |\alpha_i|^2 \leq \eta \end{cases} \tag{15}$$

where η is a random number in the interval [0,1] r_i é is the *i*-th bit of the solution *r*.

In that way, a complete set of binary solutions can be formed starting from the several possible states represented by the Q-bits. An individual *r* of QEA corresponds, in QACO, to the route traveled by one of the quantum ants between the nest and the food source, but the way it is built is a little different. In QACO the q-bit represents the pheromone. The pair of nodes '0' and '1' in figure (1) is represented by a q-bit, that is, $\tau_{j,\alpha}$ and $\tau_{j,\beta}$ of the q-bit represents the nodes '0' and '1', while $|\tau_{j,\alpha}|^2$ and $|\tau_{j,\beta}|^2$ are the probabilities that the ants will choose the path for the node '0' or '1' respectively.

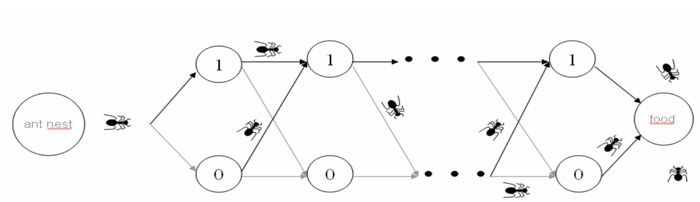

Figure 1. Binary ant colony optimization

First, a random number *p* is generated and compared with the parameter of probability of exploration *pe*. If *p* goes larger or equal to *pe*, the path of the ant *i* at the *j*-th bit is determined by the equation (16)

$$Path_{i,j} = \begin{cases} 0 & \text{if } \tau_{j,\beta} \leq \tau_{j,\alpha} \\ 1 & \text{if } \tau_{j,\beta} > \tau_{j,\alpha} \end{cases} \tag{16}$$

where $Path_{i,j,}$ is the *j*-th bit of the *i*-th ant. If *p* becomes smaller than *pe*, the route of the ant *i* at the *j*-th bit is determined by the following limiting function

$$\eta_c(x) = \begin{cases} 0 & c < \eta_0 \\ 1 & c \geq \eta_0 \end{cases} \tag{17}$$

where *c* is a random number and η_0 is a constant. From the observation of the Q-bit pheromone, a valid solution will be generated and shown in binary form by an ant walking from the nest to the food source. After the corresponding conventional solutions are generated, not only in QACO, but also in QEA ("$\tau_{j,\alpha}$, $\tau_{j,\beta}$" may be substituted by "α_{id}, β_{id}") each individual's fitness is evaluated and the q-bits updated through the quantum rotation gate $R(\theta_{id})$.

$$\begin{bmatrix} \tau_{i\alpha}^{'} \\ \tau_{i\beta}^{'} \end{bmatrix} = R(\theta_{id}) \begin{bmatrix} \tau_{i\alpha} \\ \tau_{i\beta} \end{bmatrix} = \begin{bmatrix} \cos(\theta_{id}) & -\sin(\theta_{id}) \\ \sin(\theta_{id}) & \cos(\theta_{id}) \end{bmatrix} \cdot \begin{bmatrix} \tau_{i\alpha} \\ \tau_{i\beta} \end{bmatrix} \tag{18}$$

where θ_{id} is the rotation angle, responsible for doing the individual Q-bit to converge to the fittest state ('0' or '1'). The definition of the rotation angle does not has a solid theoretical basis being defined as

$$\theta_{id} = S(\tau_{j,\alpha}, \tau_{j,\beta}).\ \Delta\theta_{id} \tag{19}$$

where $S(\tau_{j,\alpha}, \tau_{j,\beta})$ is the sign of θ_{id}, that determines the direction, and $\Delta\theta_{id}$ is the amplitude of the rotation angle. Those values are determined in QACO as given in table (1)

Table 1. Rotation angle of QACO

				$S(\tau_{i\alpha}, \tau_{i,\beta})$			
x_i	b_i	$f(x)>f(b)$	$\Delta\theta_{id}$	$\tau_{i,\alpha}\tau_{i,\beta} > 0$	$\tau_{i,\alpha}\tau_{i,\beta} < 0$	$\tau_{i,\alpha} = 0$	$\tau_{i,\beta} = 0$
0	0	False	0.01π	-1	+1	±1	±1
0	0	True	0.01π	-1	+1	±1	±1
0	1	False	0.025π	-1	+1	±1	±1
0	1	True	0.025π	+1	-1	±1	±1
1	0	False	0.025π	+1	-1	±1	±1
1	0	True	0.025π	-1	+1	±1	±1
1	1	False	0.01π	+1	-1	±1	±1
1	1	True	0.01π	+1	-1	±1	±1

where x_i is the *i*-th bit of ant *x*, b_i is it *i*-th bit of the best ant, *f(x)* is the fitness of ant *x* and *f(b)* it is the fitness of the best ant.Figure (2) below presents the original QACO pseudo code.

Initialization Phase
Atribute initial pheromone

$$\tau = \begin{bmatrix} \tau_{1\alpha} & \tau_{2\alpha} & \ldots & \tau_{m\alpha} \\ \tau_{1\beta} & \tau_{2\beta} & \ldots & \tau_{m\beta} \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & \ldots & 1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} & \ldots & 1/\sqrt{2} \end{bmatrix}$$

```
Initialization Phase
        Atribute initial pheromone
        all solutions are generated with the same probability
Route construction Phase
        Each ant travels from the nest to the food source to build a entire solution by observing the pheromone Q-bit
Loop (for each generation)
        Loop(for each ant)
                Evaluate fitness
                Loop (for each bit)
                        Update pheromone
                End bit
        End ant
End generation
Goto Loop generation
Until stoppin criteria is attained.
```

Figure 2. QACO original pseudocode

3. THE NEW QACO_Alfa ALGORITHM

Although QACO worked well for bidimensional benchmark functions, due to a premature converge behavior, it doesn't present the same performance when applied to multidimensional complex functions. According to tests presented in the section 5, the number of ants and the number of bits are parameters that most influence the behavior of the algorithm, mainly concerning its performance.

The authors [7] suggest twenty ants and a string of twelve bits to represent a real number, however, in agreement with the tests accomplished in section 5, when those values are changed QACO doesn't usually obtain good results, making the value of q-bits at certain positions of the string to be stagnated in '0' or '1', characterizing the premature convergence. That happens because in the original QACO reinforcement of same pheromone exists when the compared bits are equal ($0 \rightarrow 0$ or $1 \rightarrow 1$), what contributes to the stagnation.

To this, are made alterations in QACO to overcome the stagnation of the q-bits avoid it to converge without obtaining good results. Originally, the bit on the *j*-th position of the solution *i* is compared with the bit of the same position of the best global solution; depending on the value of the fitness, the updating angle that will be applied to the q-bit is determined. In QACO_Alfa the updating of the angle is done only when the compared bits are different, depending on the factor α_0 (a constant in the interval [0,1]) used for adjustment of the updating value according to table (2).

Table 2. Rotation Angle of QACO_Alfa

x_i	b_i	$f(x)>f(b)$	$\Delta\theta_{id}$	$S(\tau_{i,\alpha}, \tau_{i,\beta})$			
				$\tau_{i,\alpha}\tau_{i,\beta} > 0$	$\tau_{i,\alpha}\tau_{i,\beta} < 0$	$\tau_{i,\alpha} = 0$	$\tau_{i,\beta} = 0$
0	1	False	$0.025\pi * \alpha_0$	-1	+1	±1	±1
0	1	True	$0.025\pi * \alpha_0$	+1	-1	±1	±1
1	0	False	$0.025\pi * \alpha_0$	+1	-1	±1	±1
1	0	True	$0.025\pi * \alpha_0$	-1	+1	±1	±1

where x_i is the *i*-th bit of ant *x*, b_i is the *i*-th bit of the best ant, *f(x)* is the fitness of ant *x* and *f(b)* the fitness of the best ant.

Another modification of the original algorithm, also with the objective of avoiding the stagnation of the q-bits, is the use of an evaporation phase that depends on the number of zeros (ZERO(*j*)) and ones (ONE(*j*)) present in the *j*-th position of the Q-bits along the generations, in other words, for the total number of generated individuals, the numbers of zeros are counted and ones at each position of the string. Figure (3) below is an illustration of an example for a population of five individuals, each one with five bits

Individual 1	1	0	0	0	1
Individual 2	1	1	1	0	0
Individual 3	0	0	1	1	1
Individual 4	1	0	0	1	0
Individual 5	1	0	1	0	1

Figure 3. Determination of ZERO(*j*) and ONE(*j*)

where each line of the 5x5 matrix is an individual. For each column, or position *j*, the number of '0s' and '1s' are counted. The highlighted column ($j = 1$) possesses four '1' and one '0'. Therefore, ONE (1)=4 and ZERO(1)=1. With this it is possible to use the angle of evaporation $\omega(j)$ described in the table (3) whose function is to give an increment in the angle to rotate the q-bit in the contrary direction of its dominant current state, '0' or '1', with the aim to reduce its stagnating tendency.

Table 3. Evaporation Angle of QACO_Alfa

	ONE (j)>ZERO(j)	ONE (j)<ZERO(j)
	$\omega(j)$	$\omega(j)$
$\tau_{i,\alpha}\tau_{i,\beta} > 0$	$-\rho$*(0.025*π)*(sub(j)/nant)*λ	ρ*(0.025*π)*(sub(j)/nant)*λ
$\tau_{i,\alpha}\tau_{i,\beta} < 0$	ρ*(0.025*π)*(sub(j)/nant)*λ	$-\rho$*(0.025*π)*(sub(j)/nant)*λ
$\tau_{i,\alpha} = 0$	ρ*(0.025*π)*(sub(j)/nant)*λ	
$\tau_{i,\beta} = 0$		ρ*(0.025*π)*(sub(j)/nant)*λ

where ρ is a constant, sub(*j*) is the difference between the number of zeros (ZERO(j)) and ones (ONE (j)), *nants* is the number of ants considered on the problem and λ is the evaporation coefficient, a constant in the interval [0,1]. In the evaporation phase the updating is made by equation (20).

$$\begin{bmatrix} \tau'_{i\alpha} \\ \tau'_{i\beta} \end{bmatrix} = R(\omega(j)) \begin{bmatrix} \tau_{i\alpha} \\ \tau_{i\beta} \end{bmatrix} \tag{20}$$

where $R(\omega(j))$ is the rotation gate, equation (21), used in the evaporation.

$$R(\omega(j)) = \begin{bmatrix} \cos(\omega(j)) & -\sin(\omega(j)) \\ \sin(\omega(j)) & \cos(\omega(j)) \end{bmatrix} \tag{21}$$

4. TESTING FUNCTIONS

In order to test the performance of QACO_Alfa as an optimization tool, it was applied to simple benchmark functions, to a complex multidimensional and to a real problem of identification of accidents in PWR nuclear power plants.

4.1 Benchmark Functions

In order to evaluate its behavior, the QACO_Alfa was applied to the solution of some well-known benchmarks problems, the Himmeublau and Goldstein-Price functions that had also been tested in original QACO. Due to the simplicity of those functions, both algorithms didn't present problems to solve them, in other words, they were not suitable to make comparisons between them. Therefore we started to use the function $\sum x_i^2$, for which, according the tests presented in the section 5, the original QACO began to present premature convergence when n, the number of variables, was done larger than 10. The QACO_Alfa, even for n = 30, obtained the best results for this problem. The table (4) it presents the benchmarks functions used in the tests.

Table 4. Benchmark Functions

Function	x_1	x_2	Best known value
Himmeublau, range $-10 \le x_i \le 10$			
$((x_1^2)+x_2-11)^2+(x_1+(x_2^2)-7)^2$	-3.779	-3.283	0.000
	-2.805	3.131	
	3.584	-1.848	
	3.000	2.000	
Goldstein-Price, range $-2 \le x_i \le 2$			
$(1+(x_1+x_2+1)^2*(19-14*x_1+3*x_1^2-14*x_2+6*x_1*x_2+3*x_2^2))*$ $(30+(2*x_1-3*x_2)^2*(18-32*x_1+12*x_1^2+48*x_2-36*x_1*x_2+27*x_2^2))$	0.000	-1.000	3.000
n-dimension Sphere, range $-5.12 \le x_i \le 5.12$			
$\sum x_i^2$ $\quad i = 1,\ldots,n$	$x_i = 0.000$		0.000

4.2. QACO_Alfa and the Accident Identification Problem

For a nuclear power plant works normally, it is necessary to monitor a large number of variables (pressure, temperature, etc..), which means that we need to know how these variables evolve over time. This makes possible, in the deviation of the normal operation of the plant, identify the anomalous event and act to resolve it before an accident occurs. With the objective of assisting the operator of the plant to reduce their cognitive effort, thereby increasing their time to keep the power plant running in its normal security, which he has to take the best decision in due time, were created the identification of transients systems.

It is classified as a transient an anomalous event that occurs in the plant, that is, an event in which the temporal evolution of the behavior of variables which determines whether the plant operates within the limits of safety is not in accordance with its normal pattern. The problem of identification of accidents in a PWR nuclear power plant is resolved by the classification of transients.

To analyze this problem are considered the evolution of the variables against time for three accidents: Blackout - loss of external power, LOCA -loss of the coolant in the reactor primary system and SGTR - rupture of the tube in the steam generator.

The method adopted for making such a classification is based on minimizing the Euclidean distance between the transient. Each accident possesses a fingerprint that represents the temporary evolution of the involved process variables. When an event occurs, a comparison among the distances from the event to the curves of each signature is done. The event will be classified as the accident to which its distance is the closest [8].The QACO_Alfa was used as a tool to perform this task.

5. RESULTS

5.1. QACO and QACO_Alfa Results

The QACO_Alfa algorithm was applied to some benchmarks functions and its results were compared to the results obtained by an implementation of the original version of QACO, testing also the influence of the number of ants and of the string size on the results. Results of the tests are shown in tables (5) and (6).

Table 5. QACO and QACO_Alfa results

Function		**nants**	**Best results**		**Average**	
			QACO	**QACO_Alfa**	**QACO**	**QACO_Alfa**
Himmeublau		20	7.076*10^-4	2.215*10^-4	0.0158	0.0107
Goldstein Price		20	3.0001	3.0001	3.0060	3.0001
$\sum x_i^2$						
α , λ		**d** = 10, **nbits** = 120				
QACO	**QACO_Alfa**					
1	0.001	20	0.2373	1.563*10^-5	3.9110	1.563*10^-5
0.2	0.01	100	0.0141	1.563*10^-5	0.0520	1.563*10^-5
0.1	0.01	200	0.0923	1.563*10^-5	0.5723	1.563*10^-5
		d = 30, **nbits** = 360				
1	0.01	20	14.2449	0.3990	24.6711	0.5394
0.2	0.02	100	3.2277	1.4730	6.8070	2.4815
0.1	0.01	200	8.3024	0.5149	20.7499	0.9907

The above results show that QACO_Alfa is more efficient than the original, do we decided to apply the algorithm to the problem of accident identification in PWR nuclear power plants. The results are shown in table (6).

Table 6. QACO and QACO_Alfa results for accident identification

Algorithm	**nants**	**Best result**	**Average**
QACO	20	174	141.8
QACO_Alfa	20	177	176.2

The column “Best results” represents the total number of correct classifications of all time instants for all classes, remembering that the optimal solution (no classification errors) would be 183 (61 x 3 time instants). Figure (4) shows the QACO_Alfa convergence behavior showing the fitness (number of incorrect classifications) as function of the number of generations.

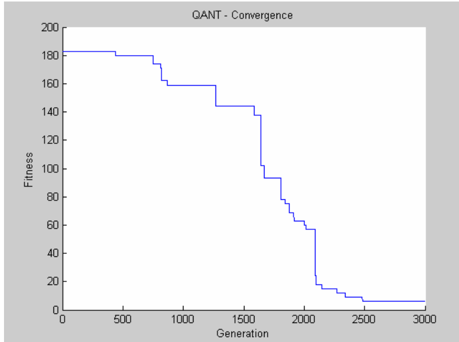

Figure 4. QACO_Alfa convergence behavior

5.2. Comparison with GA and PSO on the Identification of Nuclear Power Plant Transients

The GA and the PSO algorithms were implemented on the MATLAB environment with the aim to compare the results obtained using QACO with results of previous works [11]. For both algorithms there were performed 10 tests with different seeds for 100 generations of populations with 500 individuals. Table 8 shows GA and the PSO algorithms the number of correct classifications results for the test performed with the QACO, GA and PSO algorithms for the set of selected transients.

Table 7. Comparison of QACO, QACO_Alfa, GA and PSO results

Correct Classifications	QACO	QACO_Alfa	GA	PSO
Maximum	174	177	183	177
Minimum	9	6	3	15
Average	141.8	176.2	159.9000	160.2000

The evolution of the GA and PSO for this problem is illustrated in figures 5 (a) and (b) that present the graphic of the fitness convergence for both algorithms.

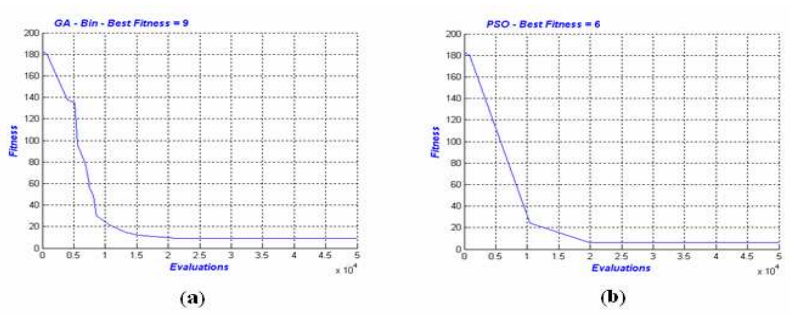

Figure 5 – (a) GA fitness convergence, (b) - PSO fitness convergence

6. CONCLUSIONS

In this work we presented an alternative version of QACO, that we denominated QACO_Alfa, to the QACO implementation proposed by Wang et al. To overcome the limitation imposed by the premature convergence in QACO, we proposed a new way to update the q-bits as well the inclusion of an evaporation stage. Our tests showed that QACO_Alfa has better performance, especially in complex multidimensional functions. Furthermore, both algorithms were applied to a real problem of identification of accidents in a PWR nuclear power plant and compared to real coded GA implementations, and once again QACO_Alfa obtained results compatible with other works, what qualifies its use as an effective tool in the resolution of similar optimization problems considered in this work.

For future works we suggest a more detailed analysis regarding the best combination of the α_0 factor and evaporation coefficient λ, as well the development of new a pseudo-heuristic with the aim to improve the performance of the algorithm.

ACKNOWLEDGMENTS

The autors are sponsored by Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq and Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro - FAPERJ

7. REFERENCES

1. Holland, J.H., “Adaptation In Natural and Artificial Systems”, Ann Arbor, *University of Michigan Press*. (1975)

2.Shor, P. W., “Algorithms for Quantum Computation: Discrete Logarithms and Factoring”, in *Proceedings of the 35th Annual Symposium on foundations of Computer Science*, pp. 124-134. (1994)

3. Han, K.H. and Kim, J.H., “Quantum-inspired evolutionary algorithm for a class of combinatorial optimization”, *IEEE Transactions on Evolutionary Computation*, **6**, pp.580-593. (2002)

4. Dorigo M. and Gambardella, L.M., Ant Colony System Tech. Rep. IRIDIA/96-01, Université Libre de Bruxelles, Belgium, EU.(1996)

5. Goss. S., Aron. S, Deneubourg J.L. and Pasteels ,J.M. “Self-organized shortcuts in the Argentine ant”, *Naturwissenschaften* **76**, 579-581. (1989)

6. Meneses et al, “Particle Swarm Optimization with random keys applied to the nuclear reactor reload problem”, *International Nuclear Atlantic Conference – INAC*, Santos, SP, Brazil, September 30 to October 5. (2007)

7. Wang, L. et al, “A Novel Quantum Ant Colony Optimization Algorithm and Its Application to Fault Diagnosis”, *Transactions of the Institute of Measurement and Control*, vol. **30**, pp.313-329. (2008)

8. Medeiros, J.A.C.C. and Schirru,R., *Identification of nuclear power plant transients using the particle swarm optimization algorithm, Annals of Nuclear Energy*, **1806**, 1-7. (2007)

9. Buriol, L. S. “Algoritmo Memético para o Prblema do Caixeiro Viajante Assimétrico como Parte de um Framework para Algoritmos Evolutivos. *Tese de Mestrado*. Campinas: DENSIS/FEE/UNICAMP. (2000)

10. Dorigo, M. and Gambardella, L.M., “Solving symmetric and asymmetric TSP’s by ant colonies”, *IEEE Conference on Evolutionary Computation (ICEC)*, May 20-22, Nagoya, Japan. (1996)

11. Waintraub, M.“Algoritmos Paralelos de Otimização por Enxame de Partículasem Problemas Nucleares”, D.Sc Thesis, Programa de Engenharia Nuclear, COPPE/UFRJ, Rio de Janeiro, Brasil. (2009)