ISSN
Number 3 | February 2013

COMPUTATIONAL INTELLIGENCE SERIES

Quantum-Inspired Genetic Algorithms applied to Ordering Combinatorial Optimization Problems

Luciano R. Silveira

Ricardo Tanscheit

Marley Vellasco

Laboratório
Inteligência Computacional Aplicada

COMPUTATIONAL INTELLIGENCE SERIES

Number 3 | February 2013

Quantum-Inspired Genetic Algorithms applied to Ordering Combinatorial Optimization Problems

Luciano R. Silveira

Ricardo Tanscheit

Marley Vellasco

CREDITS
Publisher:
MAXWELL / LAMBDA-DEE
Sistema Maxwell / Laboratório de Automação de Museus, Bibliotecas Digitais e Arquivos
http://www.maxwell.lambda.ele.puc-rio.br/
Organizer:
Marley Maria Bernardes Rebuzzi Vellasco
Cover:
Ana Cristina Costa Ribeiro

Quantum-Inspired Genetic Algorithms applied to Ordering Combinatorial Optimization Problems

Luciano R. Silveira, Ricardo Tanscheit, Marley Vellasco
Department of Electrical Engineering
Pontifical Catholic University of Rio de Janeiro
Rio de Janeiro, RJ, Brazil
lucianosilveira@globo.com, ricardo@ele.puc-rio.br, marley@ele.puc-rio.br

Abstract—**This article proposes a new algorithm based on evolutionary computation and quantum computing. It attempts to resolve ordering combinatorial optimization problems, the most well known of which is the traveling salesman problem (TSP). Classic and quantum-inspired genetic algorithms based on binary representations have been previously used to solve combinatorial optimization problems. However, for ordering combinatorial optimization problems, order-based genetic algorithms are more adequate than those with binary representation, since a specialized crossover process can be employed in order to always generate feasible solutions. Traditional order-based genetic algorithms have already been applied to ordering combinatorial optimization problems but few quantum-inspired genetic algorithms have been proposed. The algorithm presented in this paper contributes to the quantum-inspired genetic approach to solve ordering combinatorial optimization problems. The performance of the proposed algorithm is compared with one order-based genetic algorithm using uniform crossover. In all cases considered, the results obtained by applying the proposed algorithm to the TSP were better, both in terms of processing times and in terms of the quality of the solutions obtained, than those obtained with order-based genetic algorithms.**

Keywords - quantum–inspired genetic algorithms, ordering combinatorial optimization, genetic algorithms, quantum bit

I. INTRODUCTION

To overcome the challenges caused by the complexity of combinatory optimization problems, researchers have utilized techniques based on mathematical programming and on evolutionary computation, particularly genetic algorithms.

Ordering combinatorial optimization problems constitute a widely studied class of combinatorial optimization problems. In general, in an ordering combinatorial optimization problem with *n* elements, the objective is to determine which ordering of these elements exhibits the best results for a pre-established criterion. The most well-known version of this type of problem is the traveling salesman problem (TSP), the objective of which is to determine the order in which *n* cities must be visited to minimize the total distance traveled. The solution is restricted in that each city can only be visited once, except for the city in which the salesman begins and ends his sales trip.

There are countless approaches to solving the TSP based on mathematical programming [1], such as the branch and bound method and Lagrangian relaxation.

Regarding genetic algorithms, a number of specific techniques can be used to address this problem [2] [3], and these are analyzed and compared at the end of this article.

For binary combinatorial optimization problems, a new approach [4] [5] [6] has been proposed to obtain a faster converging algorithm and to lower the computational cost. This approach combines the genetic algorithm technique with a quantum bit concept, employing the quantum superposition of states, and has displayed promising results when applied to the knapsack problem, particularly when compared to solutions obtained with genetic algorithms with binary representation. The principles of quantum computing and of evolutionary computation have also been employed in optimization problems using real variables without restrictions [7] [8], and these have performed better than classic genetic algorithms. Quantum-inspired approaches have already been extended to ordering optimization problems. The method presented in [9] follows the approach proposed in [4] [5], which makes use of rotation matrices to update the quantum bits, but with a special procedure that always generate feasible solutions. The approach taken here employs a different process to generate feasible solutions and does not employ rotational matrices to update the quantum bits.

This article contains four additional sections (not including the present section). Section II explains the basic concept behind quantum bits and the representation proposed for the quantum individual in ordering problems. Section III describes the ordering quantum-inspired genetic algorithm proposed in this study, explaining the process by which solutions are obtained based on the quantum individual (the observation of the quantum individual) and how these individuals evolve over successive generations. Section IV presents the results obtained with the proposed algorithms and compares them with the traditional ordering genetic algorithm technique for the TSP, for cases that consider 27, 48, 52, and 100 cities. Finally, section V presents the conclusions of this study.

II. REPRESENTATION OF THE QUANTUM INDIVIDUAL

Quantum-inspired genetic algorithms are based on the concept of a quantum bit [4]. One quantum bit (Qbit) is the smallest amount of information stored in a quantum computer, being represented by a vector (α, β), in which α and β are complex numbers such that $|\alpha^2|$ and $|\beta^2|$ provide the probability of a quantum bit being observed in state 0 or in state 1, respectively. Naturally, the following equation must hold

$$\left|\alpha^2\right| + \left|\beta^2\right| = 1 \tag{1}$$

In genetic algorithms with binary representation, each gene of an individual (known as a classic individual) is represented by a bit. In the case of quantum-inspired genetic algorithms applied to the combinatorial optimization problems, each gene of an individual is represented by a quantum bit. The idea behind quantum-inspired genetic algorithms is to generate each classic individual based on the quantum individual. To obtain a classic gene from a quantum gene, it is sufficient to observe or randomly select the quantum bit based on the α and β values of the Qbit. The algorithm proposed in [4] to solve binary combinatorial optimization problems, applied to the Knapsack Problem, represents the quantum individuals through an n dimensional vector of Qbits. The genes of the classic individual are obtained by randomly observing each quantum gene, based on the probabilities of obtaining states 0 or 1. Assuming that the quantum individual is represented by the vector of Qbits:

$$\left(\begin{bmatrix}\alpha_1\\ \beta_1\end{bmatrix},\begin{bmatrix}\alpha_2\\ \beta_2\end{bmatrix},\ldots\ldots,\begin{bmatrix}\alpha_n\\ \beta_n\end{bmatrix}\right)$$

and that the observed classic individual is represented by the vector of bits (c_1, c_2,……, c_n), then:

$c_i = 0$ if $rnd_i \leq \beta_i^2$ and $c_i = 1$,otherwise; where rnd_i is a random value selected from the interval [0,1].

In an ordering combinatorial optimization problem with n dimensions, the individual (known as a classic individual) is represented by a vector whose n components assume values between 1 and n, with no repetitions. In the quantum version of an ordering genetic algorithm, the quantum individual that is adopted should be capable of producing classic individuals. When these concepts are applied to the TSP, the classic individual represents a sequence of visits to all cities. For each visit or stop of the salesman, a set of n Qbits is employed to define the probability of each city being selected for that visit. Assuming that the visit of interest is the second one, n Qbits will be associated to it, each of them controlling the probability of selection of each city for the second stop. Considering all stops, n^2 Qbits will be needed and a possible representation for the quantum individual for the ordering problem is an n x n matrix of quantum bits in which the j^{th} quantum bit of the i^{th} row of the matrix determines the probability of the j^{th} city being selected in the i^{th} stop of the trip. Therefore, the representation of the quantum individual is:

$$Q = \begin{bmatrix} q_{11}, q_{12}, \ldots\ldots, q_{1n} \\ q_{21}, q_{22}, \ldots\ldots, q_{2n} \\ \ldots\ldots\ldots \\ q_{n1}, q_{n2}, \ldots\ldots, q_{nn} \end{bmatrix}, \text{ in which, } q_{ij} = \left|\beta_{ij}^2\right| \tag{2}$$

To clarify, q_{35} represents the probability of city 5 being the third city to be visited.

The quantum bits that make up the quantum individual are related to one another, given that the probability of some city being selected as a particular stop should be 1 (to adhere to the restrictions of the problem). This relationship is expressed by:

$$\sum_{j=1}^{n} q_{ij} = 1, \forall i = \overline{1,n} \tag{3}$$

Once the structure of the ordering quantum individual is established, it must be defined how the process will begin, how it will be observed (to generate solutions for the problem), and how it will be updated throughout the iterations of the algorithm. These points are described in detail in the next section.

III. Ordering Quantum-Inspired Genetic Algorithms

The structure of a quantum-inspired genetic algorithm is simple and is displayed as follows:

Initialize Quantum Population
While No (Maximum Number of Generations)
    Observe Quantum Population
    Update Quantum Population
    Increase Number of Generations
End While.

The stages of the algorithm are explained in detail below.

A. Initialization of the Quantum Population

In traditional genetic algorithms the population is commonly initialized in a random manner. However, for a quantum population, a more appropriate method would be to initialize the quantum individual such that equal probabilities are ensured in generating solutions for the ordering combinatorial optimization problem. A solution to this problem is a vector of dimension n with elements that have values between 1 and n with no repetitions; this is called the classic individual because it is analogous to the individual used in a traditional ordering genetic algorithm.

According to the formulation adopted for the traveling salesman problem, the trip begins and ends in city 1, and the quantum individual should be initialized in a way that ensures an equal probability in the generation of classic individuals. Then, the initial quantum individual (or generation 0) is represented as follows:

$$Q(0) = \begin{bmatrix} 1, \quad 0 \quad, \ldots\ldots, \quad 0 \\ 0, \dfrac{1}{n-1}, \ldots\ldots, \dfrac{1}{n-1} \\ \ldots\ldots\ldots \\ 0, \dfrac{1}{n-1}, \ldots\ldots, \dfrac{1}{n-1} \end{bmatrix} \tag{4}$$

The first row of the quantum individual is always of the form (1, 0, …, 0). This is because the probability of city 1 being selected as the first stop must be 1 (by definition), and consequently, the probability of any other city being selected as the first city is 0. The other rows of the quantum individual indicate that the probability of a city being selected for any stop is always equal to $\frac{1}{n-1}$ because the probability of city 1 being selected will always be 0 in any row other than the first one. This initial quantum individual satisfies conditions (2) and (3).

B. Observation of the Quantum Population

Once a quantum individual is generated, the next action is to generate classic individuals, a process called observation of the quantum individual.

Consider, for example, the second row of the representative matrix of any quantum individual (0, $q_{22},..., q_{2n}$). As condition (3) must always be satisfied, upon choosing a random number, which has a uniform distribution in a half-open interval (0,1] called r_2, there will always be an index k, such that:

$$\sum_{j=1}^{k-1} q_{2j} < r_2 \quad \text{and} \quad \sum_{j=1}^{k} q_{2j} \geq r_2 \tag{5}$$

Therefore, city k will be the second city to be visited. To determine the third stop of the trip, the third row of the quantum individual (0, $q_{32},..., q_{3k},..., q_{3n}$) must be considered. The process performed to determine the city of the second stop cannot be directly applied to the original third row of the quantum individual when q_{3k} is nonzero, because city k can no longer be selected as the third stop (if it is equal to zero, then the third row is ready to be processed). Therefore, it is necessary to update the probabilities with which the cities may be selected for the third stop. Knowing that city k must have zero probability to be selected and that the conditions in (3) must always be satisfied, the third line of the quantum individual must be altered so that the following holds:

$$(0, \frac{q_{32}}{1-q_{3k}}, ..., 0, ..., \frac{q_{3n}}{1-q_{3k}}) \tag{6}$$

This new updated row ensures that city k cannot be selected a second time. Given that

$$\sum_{j=2}^{k-1} q_{3j} + \sum_{j=k+1}^{n} q_{3j} = 1 - q_{3k} \tag{7}$$

Then:

$$\sum_{j=2}^{k-1} \frac{q_{3j}}{1-q_{3k}} + \sum_{j=k+1}^{n} \frac{q_{3j}}{1-q_{3k}} = 1 \tag{8}$$

Thus, condition (3) is still satisfied by the new row 3. Naturally, the process described to update the third row must be applied to all rows that have yet to be processed. After rows 3, 4, …, n have been updated, the process of determining the third stop follows the same steps described to obtain the second city. Once the third stop is determined, the process of updating the unprocessed rows is repeated and the cities of the other stops are chosen until all stops have been determined.

The rows are not necessarily processed sequentially (first, second, third, etc.); any order is valid for generating the classic individual.

In short, let:

- I_P be the set of already processed rows;
- C_P be the set of the cities which have already been visited;
- $i \in \{1, 2,..., n\} - I_P$ be the next row to be processed;
- k be the city chosen in the selection process of row i.

Assuming that (0, $q'_{m2},..., q'_{mk}, ..., q'_{mn}$) is the current representation of row $m \in \{1, 2, ..., n\} - I_P$ (after all of the updates made in processing the rows belonging to I_P), then the elements of rows $m \in \{1, 2,..., n\} - I_P - \{i\}$ will be represented by q''_{mj}, $m \in \{1, 2,..., n\} - I_P - \{i\}$, $j \in \{1, 2,..., n\} - C_P - \{k\}$, in which:

$$q''_{mj} = \frac{q'_{mj}}{1-q'_{mk}} \tag{9}$$

If the observation process of the quantum individual is performed *ncla* times, then *ncla* classic individuals will be generated.

C. Updating the Quantum Population

Each classic generated individual is evaluated by the criterion function that has been established; the best classic individual obtained (with the best value in the criterion function) from each quantum individual is used to update this quantum individual. Let's assume that the algorithm is at generation t and that $Q(t)$ is a quantum individual of this generation. Let's also assume that the observation process for the quantum individual produces *ncla* classic individuals and that the best individual is represented by vector $c(t)$ (a vector with n dimensions whose elements vary between 1 and n, with no repetitions, and the first position of $c(t)$ is equal to 1). The vector $c(t)$ may then be represented as a matrix, ordering the rows of the n x n identity matrix in the order of the elements of $c(t)$. This matrix will be called $E(t)$. For example, if $c(t)$=(1, 3, 2), then matrix $E(t)$ will be formed by rows $(1, 0, 0)^T$, $(0, 0, 1)^T$, and $(0, 1, 0)^T$.

The quantum individual is updated in order to increase the probability of the observation of the classic individual $c(t)$. Thus, every other solution will have a reduced probability of observation.

One possible way of updating the quantum individual is to use the following recurrence equation:

$$Q(t+1) = (1-\varepsilon(t))Q(t) + \varepsilon(t)E(t) \tag{10}$$

In equation (10), $\varepsilon(t)$ is a parameter to be specified; it has a value between 0 and 1 and is either dependent on the generation or not.

Because it satisfies conditions (2) and (3), the quantum individual $Q(t+1)$ remains valid. Condition (2) is clearly satisfied because each element of $Q(t+1)$ is a convex combination of one element from $Q(t) \in [0,1]$ and one element from $E(t) \in \{0,1\}$. For the conditions in (3), it follows that:

$$\sum_{j=1}^{n}(1-\varepsilon(t))q_{ij}(t)+\varepsilon(t)=1-\varepsilon(t)+\varepsilon(t)=1 \tag{11}$$

It remains to be shown that the probability of observing the classic individual $c(t)$ is greater in $Q(t+1)$ than in $Q(t)$. For this, it is sufficient to verify that

$$\begin{aligned} &q_{ij}(t+1)=(1-\varepsilon(t))q_{ij}(t)\le q_{ij}(t), \forall j \ne c(i) \\ &q_{ij}(t+1)=(1-\varepsilon(t))q_{ij}(t)+\varepsilon(t)\ge q_{ij}(t), j=c(i) \\ &\forall i \in \{1,2,...,n\}, \end{aligned} \tag{12}$$

with equality occurring only in the case in which

$$q_{ij}(t)=1, j=c(i) \text{ e } q_{ij}(t)=0, \forall j \ne c(i) \ .$$

It should be noted that, because the probability of observing $c(t)$ in $Q(t+1)$ is greater than in $Q(t)$, over the course of the generations, the quantum individual may reach a situation in which any observation made always produces the same classic individual. In almost every numerical analysis that has been performed, the capacity of the quantum individual to generate distinct classic individuals is severely affected when a certain number of generations are reached. The greater coefficient for each row of the matrix (representing the quantum individual) is a good indication of the capacity of the quantum individual to generate distinct classic individuals. Let's assume that the greatest coefficient in the third row is 0.98 (in column 5). If the third row were the first one to be processed, city 5 will be selected as the third stop of the trip in 98% of the observations.

Expanding this idea, let s_i be the greatest coefficient in line i and consider that $s = \mathrm{Min}_{i=1,n}\{s_i\}$. A high value of s (greater than 0.99, for example) affirms that the expectance of observing multiple distinct classic individuals is small.

The algorithm presented in this section was tested with four different configurations of the traveling salesman problem. The next section presents the results.

IV. RESULTS

Multiple traveling salesman problems were analyzed using configurations of 27, 48, 52, and 100 cities. The configurations of 48, 52 and 100 cities can be found in TSPLIB95[1] under the names att48, berlin52 and kroc100, respectively. Each configuration was solved with an ordering genetic algorithm and with the proposed algorithm. Ten experiments were performed for each algorithm, and the minimum, mean and maximum values of the criterion function are presented for each experiment, as well as the number of evaluations of the criterion function.

For the ordering genetic algorithm, uniform order crossover was adopted, and for a problem with n cities, $50n$ generations and $2n$ classic individuals were considered. Thus $100n^2+2n$ (including initializing) evaluations of the criterion functions will be performed for each experiment.

For the quantum-inspired genetic algorithm, the numbers of generations, quantum individuals and classic individuals observed are chosen so that the number of evaluations of the criterion function is equivalent to that of the ordering genetic algorithm. The performance of the quantum inspired algorithm depends on the number of observations of each quantum individual for every generation and on the values used to update the quantum individuals. The number of observations is a parameter named *ncla* and the value used to update a quantum individual is obtained as follows. Considering a quantum individual q, each observed classic individual is evaluated by the criterion function and the best individual (the $c_q(t)$ vector) exhibits the total distance covered, which is denominated $FGer_q(t)$). Throughout the generations, the best classic individual obtained for each generation is stored, and the distance covered in this solution is called $FMin_q$. The parameter $\varepsilon_q(t)$, used to update the quantum individual, is calculated by the following equation:

$$\varepsilon_q(t)=\varepsilon_{base}\left(\frac{FMin_q}{FGer_q(t)}\right)^p \tag{13}$$

The values adopted for ε_{base} and p were parameterized. Note that the ratio $\dfrac{FMin_q}{FGer_q(t)}$ is always less than or equal to one and that the lower this value is, the worse the solution found in the observation process of the quantum individual in generation t will be. The use of (13) reduces the effect of a bad solution in updating a quantum individual (10). The parameter p works as a reducer for the updating; the greater the power p, the less the effect of a bad solution (a small $\frac{FMin_q}{FGer_q(t)}$ ratio) on updating the quantum individual.

As discussed in section III.C, the quantum individual may eventually exhaust its capacity to generate distinct classic individuals. By adopting the smallest among the large coefficients of each row as a control parameter, as long as this value is greater than 0.99, the quantum individual will stop its processing and the best solution obtained so far will be used as the one produced by the quantum individual. This control parameter will be called saturation of the quantum individual

[1] http://www2.iwr.uni-heildelberg.de/groups/comopt/software/TSPLIB95/tsp

and it is easy to verify that its values lie within $\left[\frac{1}{n-1},1\right]$. As the probability to observe distinct classic individuals decreases as saturations increases, the number of observations of the q^{th} quantum individual at generation t will be given by:

$$ncla_q(t) = \frac{n-1}{n-2}(NC-1)s_q(t)+1 \tag{14}$$

where $ncla_q(t)$ is the number of observations of the quantum individual q at generation t, $s_q(t)$ is the saturation value of the quantum individual q at generation t and NC is as parameter to be determined. Expression (14) indicates that the number of observations depends linearly on saturation; it is equal to NC if the saturation value is $\frac{1}{n-1}$ and 1 when the saturation value is 1.

The tests performed with the quantum-inspired genetic algorithm used different values for parameters NC, ε_{base} and p. Random order processing was used for the lines of the representative matrix of the quantum individual in the observation process.

In the tables in the following sections, the abbreviated terms have the following meanings:

NQ: number of quantum individuals considered;

NC: maximum number of observed classic individuals;

$\boldsymbol{\varepsilon_{base}}$: parameter (see 13);

$\boldsymbol{p}$: parameter (see 13);

Min: the minimum value obtained for the criterion function for the 10 experiments;

Avg: the mean value obtained for the criterion function for the 10 experiments;

Max: the maximum value obtained for the criterion function for the 10 experiments;

NE: the maximum number of evaluations of the criterion function for the 10 experiments.

A. *Results for 27 cities*

The 27-cities configuration refers to the problem that considers the 27 Brazilian state capital cities, beginning and ending in Rio de Janeiro. The optimal solution to this problem was obtained through the integer programming technique [10], and the smallest possible distance covered was 17,028 km. By using a ordering genetic algorithm (1350 generations and 54 individuals) the best solution (average result) for the 10 experiments was obtained for a crossover rate of 1.00, mutation rate of 0.02, and elitism of 10% (i.e. the top 10% of individuals in each generation progress to the next generation). The minimum, average and maximum solutions were 19489, 20954 and 22556, respectively. A total of 749,540 evaluations of the criterion function were performed and the processing time was less than 2 s. The results for the quantum inspired algorithm are shown in Table 1 (2700 generations and processing time about 2 s for each set of parameters).

TABLE 1. RESULTS FOR 27 CITIES

NQ	NC	ε_{base}	p	Avg	Min	Max	NEval
1	27	0.015	0	21139	18089	25568	317,015
1	27	0.015	1	20890	18153	23433	326,533
1	27	0.015	2	19994	18269	23162	364,030
1	27	0.015	3	21446	19810	24399	369,087
1	27	0.015	4	19773	17028	21781	435570
1	27	0.020	0	22531	18225	25229	210,989
1	27	0.020	1	20603	17028	23194	227,450
1	27	0.020	2	22029	17436	26707	258,430
1	**27**	**0.020**	**3**	**19648**	**17028**	**22456**	**262,666**
1	27	0.020	4	21248	19357	26778	335,125

Although, processing times are equivalent, the best average solution for the quantum inspired algorithm outperforms that for the ordering genetic algorithm by 6.23%. With the use of the quantum inspired algorithm the number of evaluations of the criterion function is reduced by a factor of 0.36. As the parameter p increases, more evaluations of the criterion function are needed because the quantum individual is updated with small values (Eq. 13). Regarding the minimum and maximum solutions, the quantum inspired algorithm provided gains of about 12.63% and 0.44%.

It should be noted that the best solution obtained by the proposed algorithm is identical to the optimal solution for this problem.

B. *Results for 48 cities*

The 48-cities problem includes traveling to 48 state capitals in the United States of America (att48). The optimal solution of this problem is a tour with length of 10,628. By using the ordering genetic algorithm (2400 generations and 96 individuals) the best average result for the 10 experiments was obtained for a crossover rate of 1.00, mutation rate of 0.01 and elitism of 10%. The minimum, average and maximum solutions were 11828, 13454 and 16280, respectively. A total of 2,304,960 evaluations of the criterion functions were performed and the processing time was less than 9 s. The results for the quantum inspired algorithm are shown in Table 2 (4800 generations and processing time of about 18 s for each set of parameters).

The processing time for the proposed algorithm is about twice that for the ordering genetic algorithm but the best average solution obtained by the quantum inspired algorithm is 5,22% better; the number of evaluations of the criterion function is reduced by a factor of 0.76. The behavior of the proposed algorithm concerning parameter p was the same as in the previous problem. It is worth mentioning the benefit obtained when two quantum individuals are used, except for the power of 4. Regarding the minimum and maximum solutions the gains were about 0.17% and 11.13% with respect to the genetic algorithm.

TABLE 2. RESULTS FOR 48 CITIES

NQ	NC	ε_{base}	p	Avg	Min	Max	NEval
1	48	0.01	0	13472	12300	14877	1,292,719
1	48	0.01	1	13993	12357	15488	1,453,491
1	48	0.01	2	13424	11687	15512	1,512,381
1	48	0.01	3	13290	12143	14052	1,685,151
1	48	0.01	4	13170	12025	14315	1,821,684
2	**24**	**0.01**	**0**	**12725**	**11808**	**14468**	**1,760,347**
2	24	0.01	1	13020	12088	14329	1,931,600
2	24	0.01	2	13000	12046	14296	2,027,459
2	24	0.01	3	12932	11863	14978	2,070,556
2	24	0.01	4	13717	12309	15222	2,109,300

The best solution (third line in Table 2) was 9.96% worse than the optimal solution for this problem.

C. *Results for 52 cities*

The 52-city problem (berlin52) considered locations in the city of Berlin. The optimal solution of this problem is a tour with length of 7,542. By using the ordering genetic algorithm (2600 generations and 104 individuals) the best average result for the 10 experiments was obtained for a crossover rate of 1.00, mutation rate of 0.03 and elitism of 10%. The minimum, average and maximum solutions were 9068, 9907 and 10684, respectively. A total of 2,653,020 evaluations of the criterion function were performed and the processing time was less than 11 s. The results for the quantum inspired algorithm are shown in Table 3 (5200 generations and processing time of about 22 s for each set of parameters).

TABLE 3. RESULTS FOR 52 CITIES

NQ	NC	ε_{base}	p	Avg	Min	Max	NEval
1	52	0.01	0	9616	8728	10023	1,498,121
1	52	0.01	1	9571	8931	10548	1,569,538
1	52	0.01	2	9495	8898	10224	1,699,698
1	52	0.01	3	9737	9218	10122	1,878,582
1	52	0.01	4	9465	8711	9949	1,987,600
2	26	0.01	0	9428	8812	10445	1,992,334
2	26	0.01	1	9232	8617	9858	2,125,711
2	**26**	**0.01**	**2**	**9143**	**8678**	**9622**	**2,207,771**
2	26	0.01	3	9222	8758	9817	2,353,256
2	26	0.01	4	9329	8752	10009	2,425,091

Processing time for the proposed algorithm is again about twice that for the genetic algorithm, but the best average solution is 7.71% better, and the number of evaluations of the criterion function is reduced by a factor of 0.83. The results are similar to those obtained for 48 cities. The use of two quantum individuals showed to be better for all powers. Regarding the minimum and maximum solutions the gains were about 4.30% and 9.94%.

The proposed algorithm's best solution (seventh line in Table 3) is 14.25% worse than the optimal one for this problem.

D. *Results for 100 cities*

This last configuration considers 100 locations (kroc100). The optimal solution is a tour with length of 20,749. By using the ordering genetic algorithm (5000 generations and 200 individuals) the best average result for the 10 experiments was obtained for a crossover rate of 1.00, mutation rate of 0.02 and elitism of 10%. The minimum, average and maximum solutions were 24298, 28274 and 34956, respectively. The criterion function was evaluated 10,002,000 times and the processing time was less than 85 s.

For this problem a slight modification was implemented. Equation (14) is used as long as the number of generations is below 2/3 of the total of generations. When the number of generations is beyond this limit, the parameter NC is doubled. The objective of this modification is to guarantee that the quantum individuals reach the state of saturation before the generations finish. When multiple quantum individuals are employed (more than three) a rule to force them to change information is applied. After 10% of the total number of generations, the quantum individual that observed the worst solution (based on the criterion function) is updated, at each generation, by using the best solution observed among all quantum individuals (and not by its own solution). Results for the quantum inspired algorithm are shown in Table 4 (30000 generations and processing time of about 300 s).

Processing time for the proposed algorithm, when compared to the genetic algorithm, is almost four times larger (some extra processing time was needed in order to prevent numerical instability). The best average solution from quantum inspired algorithm is 6.20% better, and the number of evaluations of the criterion function is reduced by a factor of 0.79. The behavior of the proposed algorithm concerning the parameter p is different from the previous problems. The number of evaluations of the criterion function decreases as long as the power increases (except for power 4). Regarding the minimum and maximum solutions the gains were about -1.32% and 18.63% with respect to the genetic algorithm.

The proposed algorithm's best solution (first line in Table 4) is 14.49% worse than the optimal solution for this problem.

TABLE 4. RESULTS FOR 100 CITIES

NQ	NC	ε_{base}	p	Avg	Min	Max	NEval
4	10	0.05	0	28274	23963	31687	6,471,566
4	10	0.05	1	27892	25943	29580	6,394,866
4	10	0.05	2	29155	24516	31595	5,506,115
4	10	0.05	3	27875	26425	30963	5,079,966
4	10	0.05	4	29579	27513	33945	7,306,298
5	10	0.05	0	28994	26234	32921	9,408,336
5	**10**	**0.05**	**1**	**26521**	**24619**	**28443**	**7,894,853**
5	10	0.05	2	28605	26136	30617	9,337,997
5	10	0.05	3	28953	26827	31112	9,869,944
5	10	0.05	4	29921	25426	51165	10,031,647

V. CONCLUSION

Quantum-inspired genetic approaches have been used to address ordering combinatorial optimization problems by using the standard rotation matrices method to update the quantum bits. A new approach has been presented in this paper and was applied to the well known Traveling Salesman Problem as a case study. Four instances with different sizes were considered, and the results obtained suggest that the performance of a quantum-inspired genetic algorithm (for solving ordering problems) is superior to that of genetic ordering algorithms that employ uniform crossover. Further research, such as crossover and mutation of quantum individuals, as well as the search for additional numerical results, will be necessary to establish a definitive verdict.

REFERENCES

[1] G. L. Nemhauser and L. A. Wosley, Integer and Combinatorial Optimization. Wiley Interscience Series in Discrete Mathematics and Optimization, 1988.

[2] M. Gen and R. Cheng, Genetic Algorithms and Engineering Design. Wiley Series in Engineering Design and Automation, 1997.

[3] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer, 1994.

[4] K. Han and J. H. Kim, “Genetic quantum algorithm and its application to combinatorial optimization problem,” Congress on Evolutionary Computation, pp. 1354-1360, 2000.

[5] K. H. Han and J. H. Kim, “Quantum-inspired evolutionary algorithm for a class of combinatorial optimization,” IEEE Transactions on Evolutionary Computation, vol. 6, pp. 580–593, 2002.

[6] M. D. Platel, S. Schliebs, and N. Kasabov. “A versatile quantum-inspired evolutionary algorithm,” Congress on Evolutionary Computation, pp. 423-430, 2007.

[7] A. V .A Cruz, M. B. R Vellasco, and M. A. C Pacheco, “Quantum-inspired evolutionary algorithm for numerical optimization," Congress on Evolutionary Computation, pp. 2630–2637, 2006.

[8] M. B. R. Vellasco, A. V. A. Cruz, and A. G. Pinho, “Quantum-inspired evolutionary algorithms applied to neural modeling,” IEEE World Conference on Computational Inteligence, Plenary and Invited Lectures, pp. 125-150, 2010.

[9] H. Talbi, A. Draa, M. Batouche, “A New Quantum-Inspired Genetic Algorithm for Solving the Travelling Salesman Problem.” IEEE International Conference on Industrial Technology (ICIT) 2004.

[10] H. M. Salkin, Integer Programming, Addison-Wesley Publishing Company, 1975.