Quantum Dynamic Mechanism-based Parallel Ant Colony Optimization Algorithm

Xiao-ming You *

College of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 200065, China

Sheng Liu , Yu-ming Wang

School of Management, Shanghai University of Engineering Science, Shanghai 200065, China
E-mail: {yxm6301,ls6601}@163.com
www.sues.edu.cn

Abstract

A novel Parallel Ant Colony Optimization Algorithm based on Quantum dynamic mechanism for traveling salesman problem (PQACO) is proposed. The use of the improved 3-opt operator provides this methodology with superior local search ability; several antibody diversification schemes were incorporated into the PQACO in order to improve the balance between exploitation and exploration. We describe the quantum dynamic mechanism and analysis the technology of improving performance, the efficiency of the approach has been illustrated by applying to TSP benchmark instances Chn144.

Keywords: Quantum Dynamic Mechanism, Ant system, Self-adaptive strategy, Parallelization, TSP optimization.

1. Introduction

Many application of engineering are concerned with Combinatorial Optimization (CO) problems. In the last decade, Ant Colony Optimization (ACO) has been successfully applied to many hard combinatorial optimization problems[8] and its performance has also been verified. The literature on the implementation issues as well as theoretical analysis has been steadily growing. The basic idea of ACO algorithms comes from the ability of ants to find shortest paths from their nest to food locations. It evaluates quantity and quality of the food and deposits a chemical pheromone trail on the ground which will guide other ants to the food source. Hence, cooperation between ants via the pheromone trails allows them to find the shortest paths to the food source by swarm intelligence. This characteristic of real ant colonies is exploited in artificial ant colonies to tackle optimization problems. It has drawn more and more attention of researchers, and has been widely used in varieties applications, including traveling salesman problem.

Now ACO algorithm[7, 8, 9] has been a hotspot in the research of intelligent optimization algorithms, there are two main algorithmic mechanisms involved in ACO, i.e., the selection strategies and the pheromone update mechanisms. The former guides ants to construct solutions probabilistically, and the latter helps ants to accumulate knowledge about the problem being solved, so that the search directions of ants can be biased toward promising solution regions.

ACO can usually converge to an almost optimal result in a certain number of iterations, but it becomes more difficult to speedup the algorithm when the complexity

*Corresponding author. E-mail address:yxm6301@163.com .This work was supported by National Natural Science Foundation of Shanghai (Grant No.09ZR1420800), National Natural Science Foundation of China (Grant No.61075115), and Development Foundation of SUES (Grant No.2008XY18).

Published by Atlantis Press

of the problem increases. It is feasible to parallelize the ACO into a parallel ant colony algorithm. The design of the parallel model is intended to reduce the communications, to this purpose, the following options have been chosen:

- Migration policy: how to control and to manage the information exchange between the sub colonies.
- Migration Criteria: when the information exchange between the sub-ant colonies should take place.

Dorigo M advanced a parallel ant colony algorithm on the hyper-cube architecture by modifying the rule of updating the pheromone so as to limit the pheromone values within the range of [0,1]. This new approach enhances the ability of the ant colony algorithm to deal with complicated objective functions theoretically and practically.

Quantum computing[1] is a very attractive research area. Due to its unique computational performance, the quantum computing has attracted extensive attention of researchers. It has been used to solve various Combinatorial Optimization (CO) problems[4, 5].

Some studies have used quantum ACO algorithm in solving TSP. Quantum evolutionary algorithm takes advantages of a dynamic balance between diversification and intensification. Diversification means that the search points should search globally without convergence to local minima. Intensification means that the search points should search intensively around the good solution. Proper combination of diversification and intensification is important for TSP. It can be equipped with additional mechanisms and operators which improve the quality of obtained results.

This paper presents a Parallel Ant Colony Optimization based on Quantum mechanism for TSP optimization (PQACO). By using self-organizing co-evolution strategy each subpopulation can obtain more optimal solutions. Meanwhile, because of the quantum dynamic mechanism all the subpopulations may move concurrently in a force-field until all of them reach their equilibrium states. The diversification strategy is the Q-gates Update operator. The intensification strategies are the improved 3-opt operator and the self-organizing co-evolution. It has more ability to control convergence direction by quantum measuring operator. The experiment results show that PQACO is very efficient for the TSP optimization; usually it can obtain optimal solution faster and the solution quality is improved with self-organizing co-evolution operator. We describe the technique for improving the performance of PQACO. The rest of this paper is organized as follows: Section 2 presents the quantum computing and related work. Section 3 gives the structure of PQACO and analyzes the improving operator. The experimental study is illustrated by solving CH144 in section 4. Finally, section 5 concludes with some brief remarks and future work.

2. Quantum Computing and Related Background

In quantum computing, quantum bit (Q-bit)[1] is the smallest unit of information stored in a two-state quantum computer. The characteristic of the Q-bit representation is that any linear superposition of solutions can be represented. This superposition can be expressed as follows:

$$\Psi = \alpha|0>+\beta|1>,$$

(α, β) is a pair of complex invariables, called probability amplitude of Q-bit, which satisfies $|\alpha|^2 + |\beta|^2 = 1$. Evolutionary computing with Q-bit representation has a better characteristic of population diversity than other representation, since it can represent linear superposition of state's probabilities. And quantum system of m Q-bits representation is defined as:

$$q = \left(\begin{array}{c|c|c|c} \alpha_1 & \alpha_2 & \ldots\alpha_j\ldots & \alpha_m \\ \beta_1 & \beta_2 & \ldots.\beta_j.. & \beta_m \end{array} \right)$$

Where $|\alpha_i|^2+|\beta_i|^2=1$, $i=1,2\ldots\ldots m$.

Quantum gate (Q-gate) U (t) is a variable operator of QEA, it can be chosen according to the problem. A common rotation gate used in QEA is as follows[3]:

$$G(\theta) = \begin{bmatrix} COS(\theta) & -SIN(\theta) \\ SIN(\theta) & COS(\theta) \end{bmatrix}$$

where θ represent the rotation angle.

In quantum systems, the computational space increases exponentially with the size of the system which enables exponential parallelism. This parallelism could lead to quantum algorithms exponentially faster than possible classically.

In classical computing, the possible states of a system of n bits form a vector space of n dimensions, i.e.

we have 2^n possible states. However, in a quantum system of n Q-bits the resulting state space has 2^n dimensions. It is this exponential growth of the state space with the number of particles that suggests a possible exponential speed-up of computation on quantum computers over classical computers. Each quantum operation will deal with all the states present within the superposition in parallel.

The first ACO algorithm called Ant System applied to Traveling Salesman Problem (TSP) by Dorigo. It makes up the main framework of other ACO algorithms and is considered as a prototype. In TSP each of m artificial ants generates a complete tour by a probabilistic rule Eq.(1), which is the probability that ant k in city *i* visits city *j*.

$$p_{i,j}^{k} = \begin{cases} \dfrac{\tau_{i,j}^{\alpha} \cdot \eta_{i,j}^{\beta}}{\sum_{j \in N_i^k} \tau_{i,j}^{\alpha} \cdot \eta_{i,j}^{\beta}} & j \in N_i^k \\ 0 & else \end{cases} \tag{1}$$

Where $\tau_{i,j}$ is pheromone, $\eta_{i,j}$ is heuristic function and is equal to $\frac{1}{d_{i,j}}$, the inverse of the distance between city *i* and *j*. N_i^k is the set of cities that haven't been visited by ant *k*, α and β are parameters which shows the relative importance of pheromone versus heuristic or exploitation versus exploration.

Eq.(1) shows that ants prefer paths with shorter length and higher amount of pheromone, so they independently generate tours by pre-knowledge of the problem and cooperative informative communication. Once all the ants complete their tours the pheromone trails updates, using Eq. (2) and Eq. (3).

$$\tau_{i,j} = (1-\rho)\tau_{i,j} + \sum_{k=1}^{m} \Delta\tau_{i,j}^{k} \tag{2}$$

$$\Delta\tau_{i,j}^{k} = \begin{cases} \dfrac{Q}{L_k} & the\ ant\ k\ passes\ link\ (i,j) \\ 0 & else \end{cases} \tag{3}$$

Where ρ is evaporation rate, L_k is the length of tour taken by ant *k*, *Q* is a constant, and *m* is the number of ants.

Pheromone evaporation is a process of decreasing the intensities of pheromone trails over time. This process is used to avoid local convergence and to explore more search space.

We use quantum dynamic mechanism, denoting particle S_i to be the *i*-th subpopulation in Fig.1, where the particle S_i in a force-field corresponds to the entry S_i in the matrix S=($U(S_i)$, ζ_i^j), $U(S_i)$ *is* utility of particle S_i; ζ_i^j is the intention strength between S_i and S_j . A particle may be driven by several kinds of forces[2]. The gravitational force produced by the force-field tries to drive a particle to move towards boundaries of the force-field, which embodies the tendency that a particle pursues maximizing the aggregate benefit of systems. The pushing or pulling forces produced by other particles are used to embody social cooperation or competition. The self-driving force produced by a particle itself represents autonomy of individual. Under the exertion of resultant forces, all the particles may move concurrently in a force-field until all particles reach their equilibrium states.

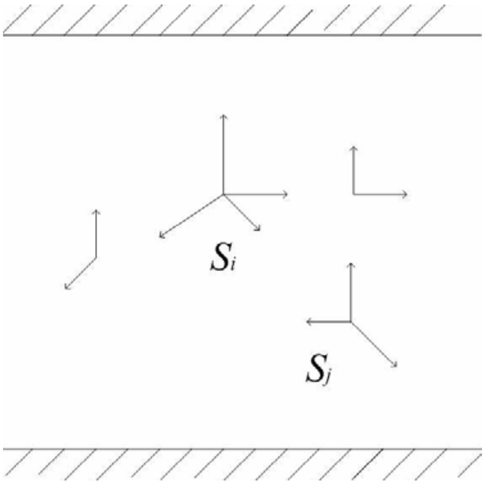

Figure 1. Particles and force-field

Given an optimization problem:

$$\begin{cases} \min f(X) \\ st.g(X) \end{cases}$$

Let U_{i} (t) be the utility of particle S_{i} at time *t*, and let *J*(t) be the aggregate utility of all particles. They are defined by Eq.(4).

$$fitness(t) = \exp(f(X))$$

$$U_i(t) = 1 - \exp(-fitness(t));$$

$$J(t) = \alpha \sum_{i=1}^{n} U_i(t), \quad 0 \prec \alpha \prec 1; \tag{4}$$

At time *t*, the potential energy function *Q*(t) that is related to interactive forces among particles is defined by Eq.(5).

$$Q_i(t) = \sum_j \int_0^{U_i(t)} \{[1+\exp(-\zeta_i^j x)]^{-1} - 0.5\} dx;$$

$$Q_D(t) = \sum_i Q_i(t) \tag{5}$$

ζ_i^j is the intention strength, which is used to describe the effect of cooperation(ζ_i^j<0) or competition(ζ_i^j>0). At time *t*, the potential energy function *P*(t) that is related to the gravitational force of force-field *F* is defined by Eq. (6).

$$P(t) = \varepsilon^2 \ln((\sum_i \exp(-U_i(t)^2/(2\times\varepsilon^2)))/n), 0 \prec \varepsilon \prec 1 \tag{6}$$

Let S_i(t) be the vertical coordinate of particle S_i at time t. The dynamic equation for particle S_i is defined by Eq. (7) and (8)

$$\begin{aligned} \frac{dS_i(t)}{dt} &= \lambda 1 \frac{dU_i(t)}{dt} + \lambda 2 \frac{dJ_D(t)}{dt} - \lambda 3 \frac{dP(t)}{dt} - \lambda 4 \frac{dQ_D(t)}{dt} \\ &= (\lambda 1 + \lambda 2 \frac{\partial J_D(t)}{\partial U_i(t)} - \lambda 3 \frac{\partial P(t)}{\partial U_i(t)} - \lambda 4 \frac{\partial Q_D(t)}{\partial U_i(t)}) \frac{dU_i(t)}{dt} \\ &= g_1(S_i(t), U_i(t)), \qquad 0 \le \lambda 1, \lambda 2, \lambda 3, \lambda 4 \le 1 \end{aligned} \tag{7}$$

$$\begin{aligned} \frac{dU_i(t)}{dt} &= \exp(-fitness(t)) \times fitness^{'}(t) \\ &= (1 - U_i(t)) \times fitness^{'}(t) \\ &= g_2(S_i(t), U_i(t)) \end{aligned} \tag{8}$$

g_2, g_1 are functions of S_i (t) and U_i(t).

If all particles reach their equilibrium states at time *t*, then finish with success.

3. Quantum Dynamic Mechanism-based Parallel Ant Colony Optimization Algorithm

Conventional quantum evolutionary algorithm (QEA) is efficacious, in which the probability amplitude of Q-bit was used for the first time to encode the chromosome and the quantum rotation gate was used to implement the evolving of population. Quantum evolutionary algorithm has the advantage of using a small population size and the relatively smaller iterations number to have acceptable solution.

Dorigo M advanced a parallel ant colony algorithm on the hyper-cube architecture by modifying the rule of updating the pheromone so as to limit the pheromone values within the range of [0,1].

Parallel Evolutionary Algorithms (PEA)[10,11,12] can be classified into three different models:

- Master-slaves PEA.
- Fine-grained PEA.
- Coarse-grained PEA.

In coarse-grained model, the population is divided into several subpopulations; each subpopulation applies the classical EA process independently with its own parameters. The design of the parallel model is intended to reduce the communications, to this purpose, the following options have been chosen:

- Migration policy:
  To reduce communications, migrations do not take place from one subpopulation to any other, but only to the next one (Fig.2.). The N cooperative subpopulations thus form a ring.

- Migration Criteria:
  Populations can send and receive individuals to and from the rest of the populations. They are randomly chosen with a probability proportional to their fitness.

To reduce communications, we improve the coarse-grained model, in which information among universes is exchanged by adopting emigration based on the learning mechanism and interaction simulating entanglement of quantum, and propose an easily expandable architecture (Fig.2.). The proposed topology for parallel architecture is basically organized with coarse-grained algorithm and each coarse-grained PEA is connected to other coarse-grained PEA group like ring topology.

Considering of the bottleneck of transferring data between Master and Slaver, the number of individuals per subpopulation is limited, if the number of Master Subpopulations is large, we organized Master subpopulations as another coarse-grained PEA, it is called hierarchical ring model (Fig.3). The Master subpopulation is only virtual host server, it is a population evolving independently, but it has function managing and collecting the best results.

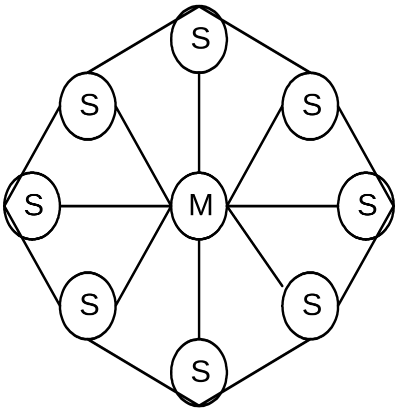

Figure2. Coarse-grained model

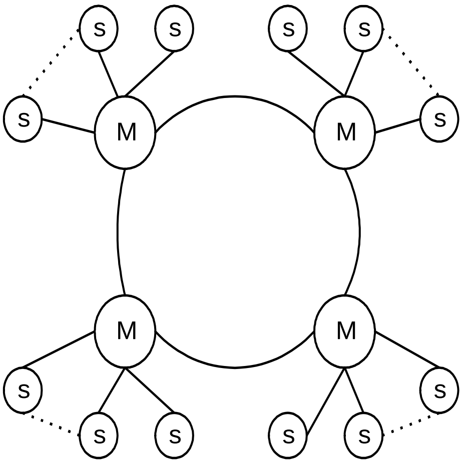

Figure 3.Improved coarse-grained model

3.1. *Migration Operator and Learning Operator between the Universes*

Parallel Evolutionary Algorithms performance is based on the interchange of individuals among different sub-populations. Here we adopt modified migration mechanism based on learning mechanism, namely, exchanging information between Slave and Master population, as show in Fig.4.

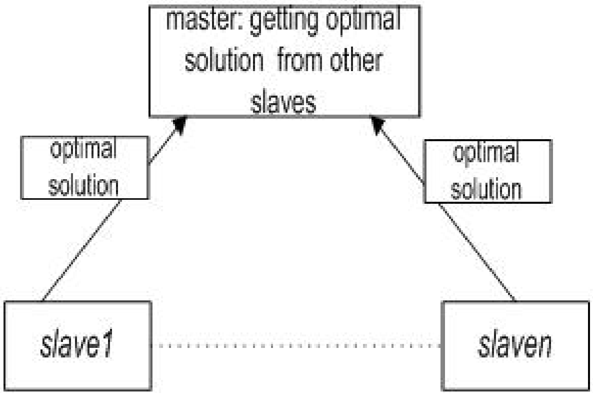

Figure 4. igration policy based on learning mechanism

Learning operator between Slave and Master population is as follows:

(i) P_s is learning frequency of Slave population and is calculated by Eq. (9):

$$p_s = \mu \frac{maxfitness\ -\ avefitness}{maxfitness\ -\ minfitness} \tag{9}$$

μ is coefficient, it is 1.0 in general.

(ii) Generating random R in [0, 1];
If $R \geq p_s$, sending optimal solution of Slave population to Master

3.2. *Population Interaction*

We proposed that information among the universes is exchanged by quantum interaction simulating entanglement of quantum, it is as follows:

(i) Randomly choose individuals from the universe with a probability proportional to their average fitness;

(ii) Update individuals chosen by step (i) using Q-gate U(t) of neighborhood universe respectively, in which information among the universes is exchanged;

(iii) Repeat (i) and (ii), until all universes complete interaction.

3.3. *Quantum Dynamic Mechanism-based Parallel Ant Colony Optimization Algorithm*

The main steps of quantum dynamic mechanism-based Parallel Ant Colony Optimization for TSP optimization (PQCAO) are as follows:

Step 1: Initialize subpopulations $Q_k(0)$ of the ant colony (m ants, n cities):

$$Q_k(0) = \left\{ q_{k1}^0, q_{k2}^0, q_{k3}^0, \ldots, q_{kn}^0 \right\}, k = 1, 2, \ldots m$$

$$q_{k\,i}^0 =$$

$$\begin{bmatrix} \begin{bmatrix} \alpha_{ki}^0(1,1) \\ \beta_{ki}^0(1,1) \end{bmatrix} & \begin{bmatrix} \alpha_{ki}^0(1,2) \\ \beta_{ki}^0(1,2) \end{bmatrix} & \cdots & \begin{bmatrix} \alpha_{ki}^0(1,n) \\ \beta_{ki}^0(1,n) \end{bmatrix} \\ \begin{bmatrix} \alpha_{ki}^0(2,1) \\ \beta_{ki}^0(2,1) \end{bmatrix} & \begin{bmatrix} \alpha_{ki}^0(2,2) \\ \beta_{ki}^0(2,2) \end{bmatrix} & \cdots & \begin{bmatrix} \alpha_{ki}^0(2,n) \\ \beta_{ki}^0(2,n) \end{bmatrix} \\ \cdots & \cdots & \cdots & \cdots \\ \begin{bmatrix} \alpha_{ki}^0(n,1) \\ \beta_{ki}^0(n,1) \end{bmatrix} & \begin{bmatrix} \alpha_{ki}^0(n,2) \\ \beta_{ki}^0(n,2) \end{bmatrix} & \cdots & \begin{bmatrix} \alpha_{ki}^0(n,n) \\ \beta_{ki}^0(n,n) \end{bmatrix} \end{bmatrix}_{n \times n}$$

Denoting $d[l,h]$ to be distance of city (i, j).

$$\beta_{ki}^0(l,h) = \begin{cases} \dfrac{d[l,h] - E_m}{L_m} \times 0.5 + 0.5 & l \neq h \wedge L_m \neq 0 \\ 0.5 \quad l \neq h \wedge L_m = 0 \end{cases}$$

$$\alpha_{ki}^0(l,h) = \sqrt{1 - (\beta_{ki}^0(l,h))^2}$$

$$E_m = \frac{1}{n} \sum_{j=1}^{n} d[m,j]$$

$$L_m = \max\left\{ \left| d[m,j] - E_m \right|, j = 1,2 \ldots n \right\}$$

In matrix q_{ki}^t , for each row and each column there is only one element which value is 1, the rest will be 0.

Let N={1,2,…,n},

for l=1 to n

h= $INT(Rand[0,1] \times n) + 1$

if $\left| \alpha_{ki}^t(l,h)^2 \right| \leq q_0$ and $h \in$ N, ($q_0 \in$[0,1])

then $\begin{bmatrix} \alpha_{ki}^t(l,h) \\ \beta_{ki}^t(l,h) \end{bmatrix}$=1; *N=N-{h}*

else

$$b = \arg\max\left(\frac{\tau_{l,j}^{\alpha}\cdot\eta_{l,j}^{\beta}}{\sum_{j\in N_i^k}\tau_{l,j}^{\alpha}\cdot\eta_{l,j}^{\beta}}\right) \qquad (b\in N)$$

$\begin{bmatrix} \alpha_{ki}^t(l,b) \\ \beta_{ki}^t(l,b) \end{bmatrix}$=1,

N=N-{b}.

End if
End for

Step 2: A Master population was selected as the virtual host server;

Other colonies are the Slave populations.

Step 3: Begin loop

For all $Q_i(t) = \left\{q_{i1}^t, q_{i2}^t, q_{i3}^t, \ldots, q_{in}^t\right\}$

{Optimize $Q_i(t)$ by 3-optlocal search algorithm;

Update $Q_i(t) = \left\{q_{i1}^t, q_{i2}^t, q_{i3}^t, \ldots, q_{in}^t\right\}$ using Q-gates $G_i(\theta)$:(θ=0.01π);

Pheromone update mechanisms;

Measure $Q_i(t)$;

Evaluate $Q_i(t)$;

Store the optimal solutions among $Q_i(t)$;

t=t+1}

Step 4: Calculating fitness of individuals, obtaining P_s (learning frequency of population) by Eq.(9);

Migrating individuals based on the learning operator; exchanging information among the subpopulation based on population Interaction.
Quantum dynamic equilibrium states for particle S_i(t) ;

Step 5: Each subpopulation can co-evolve by quantum dynamic mechanism:

If (all S$_i$(t) reach their equilibrium states)
    Finish with success;
Else
    go to **Step 3**

End.

3.4. *Update and Measure operator*

Updating $Q_i(t) = \left\{q_{i1}^t, q_{i2}^t, q_{i3}^t, \ldots, q_{in}^t\right\}$ using Q-gates G$_i$(θ) (θ=0.01π) is as follows:

$$\begin{aligned} & Q_k(t+1) \\ & = \begin{vmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{vmatrix} Q_k(t) \\ & = G_i(\theta)\bullet Q_k(t) \\ & = \left\{G_i(\theta)q_{k1}^t, G_i(\theta)q_{k2}^t, G_i(\theta)q_{k3}^t, \ldots, G_i(\theta)q_{kn}^t\right\} \end{aligned}$$

$$G_i(\theta)q_{ki}^t =$$
$$\begin{bmatrix} G_i(\theta)\begin{bmatrix}\alpha_{ki}^t(1,1)\\ \beta_{ki}^t(1,1)\end{bmatrix} & G_i(\theta)\begin{bmatrix}\alpha_{ki}^t(1,2)\\ \beta_{ki}^t(1,2)\end{bmatrix} & \ldots & G_i(\theta)\begin{bmatrix}\alpha_{ki}^t(1,n)\\ \beta_{ki}^t(1,n)\end{bmatrix} \\ G_i(\theta)\begin{bmatrix}\alpha_{ki}^t(2,1)\\ \beta_{ki}^t(2,1)\end{bmatrix} & G_i(\theta)\begin{bmatrix}\alpha_{ki}^t(2,2)\\ \beta_{ki}^t(2,2)\end{bmatrix} & \ldots & G_i(\theta)\begin{bmatrix}\alpha_{ki}^t(2,n)\\ \beta_{ki}^t(2,n)\end{bmatrix} \\ \ldots & \ldots & \ldots & \ldots \\ G_i(\theta)\begin{bmatrix}\alpha_{ki}^t(n,1)\\ \beta_{ki}^t(n,1)\end{bmatrix} & G_i(\theta)\begin{bmatrix}\alpha_{ki}^t(n,2)\\ \beta_{ki}^t(n,2)\end{bmatrix} & \ldots & G_i(\theta)\begin{bmatrix}\alpha_{ki}^t(n,n)\\ \beta_{ki}^t(n,n)\end{bmatrix} \end{bmatrix}$$

Measuring $Q_i(t)$ is as follows:

$$q_{ki}^{t+1} =$$

$$\begin{bmatrix} \begin{bmatrix}\alpha_{ki}^{t+1}(1,1)\\ \beta_{ki}^{t+!}(1,1)\end{bmatrix} & \begin{bmatrix}\alpha_{ki}^{t+1}(1,2)\\ \beta_{ki}^{t+1}(1,2)\end{bmatrix} & \cdots & \begin{bmatrix}\alpha_{ki}^{t+1}(1,n)\\ \beta_{ki}^{t+1}(1,n)\end{bmatrix} \\ \begin{bmatrix}\alpha_{ki}^{t+1}(2,1)\\ \beta_{ki}^{t+1}(2,1)\end{bmatrix} & \begin{bmatrix}\alpha_{ki}^{t+1}(2,2)\\ \beta_{ki}^{t+1}(2,2)\end{bmatrix} & \cdots & \begin{bmatrix}\alpha_{ki}^{t+1}(2,n)\\ \beta_{ki}^{t+1}(2,n)\end{bmatrix} \\ \cdots & \cdots & \cdots & \cdots \\ \begin{bmatrix}\alpha_{ki}^{t+1}(n,1)\\ \beta_{ki}^{t+1}(n,1)\end{bmatrix} & \begin{bmatrix}\alpha_{ki}^{t+1}(n,2)\\ \beta_{ki}^{t+1}(n,2)\end{bmatrix} & \cdots & \begin{bmatrix}\alpha_{ki}^{t+1}(n,n)\\ \beta_{ki}^{t+1}(n,n)\end{bmatrix} \end{bmatrix}$$

$$\begin{bmatrix}\alpha_{ki}^{t+1}(l,h)\\ \beta_{ki}^{t+!}(l,h)\end{bmatrix} =$$

$$\begin{vmatrix}\cos(\theta) & -\sin(\theta)\\ \sin(\theta) & \cos(\theta)\end{vmatrix} \begin{bmatrix}\alpha_{ki}^{t}(l,h)\\ \beta_{ki}^{t}(l,h)\end{bmatrix} 1 \le l,h \le n$$

```
let N={1,2,…n},

for $l$=1 to n

    $h= INT(Rand[0,1]\times n)+1$

    if $\left|\alpha_{ki}^{t+1}(l,h)^2\right| \le q_0$ and $h \in$ N, ($q_0 \in$[0,1])

    then $\begin{bmatrix}\alpha_{ki}^{t+1}(l,h)\\ \beta_{ki}^{t+1}(l,h)\end{bmatrix}$=1; N=N-{h}

    else

        $b = \arg\max(\frac{\tau_{l,j}^{\alpha} \cdot \eta_{l,j}^{\beta}}{\sum_{j \in N_i^k} \tau_{l,j}^{\alpha} \cdot \eta_{l,j}^{\beta}})$    ($b \in N$)

        $\begin{bmatrix}\alpha_{ki}^{t+1}(l,b)\\ \beta_{ki}^{t+1}(l,b)\end{bmatrix}$=1;

        N=N-{b}.

    End if
End for
```

Therefore, in matrix q_{ki}^{t+1} , we have only one element be equal to 1 for each row and each column, others be equal to 0.

3.5. *Self-adaptive operator and performance analysis*

Self-adaptive operator plays the key role in evolutionary algorithm generally. In matrix q_{ki}^{t} , for each row and each column there is only one element which value is 1, the rest will be 0.

```
if $\left|\alpha_{ki}^{t}(l,h)^2\right| \le q_0$ and $h \in$ N, ($q_0 \in$[0,1])

then $\begin{bmatrix}\alpha_{ki}^{t}(l,h)\\ \beta_{ki}^{t}(l,h)\end{bmatrix}$=1; N=N-{h}

else

    $b = \arg\max(\frac{\tau_{l,j}^{\alpha} \cdot \eta_{l,j}^{\beta}}{\sum_{j \in N_i^k} \tau_{l,j}^{\alpha} \cdot \eta_{l,j}^{\beta}})$    ($b \in N$)

    $\begin{bmatrix}\alpha_{ki}^{t}(l,b)\\ \beta_{ki}^{t}(l,b)\end{bmatrix}$=1,

    N=N-{b}.

End if
```

.

It could be viewed as a self-adaptive operator. We can set variable q_0: in prophase we use smaller q_0 to explore more search space and to collect useful global information, otherwise in anaphase we use bigger q_0 to accelerate convergence. This mechanism offers the ability to escape from local optima and can self-regulate the production of diverse antibodies. At first we set $q_0 = 0.3(or0.4)$, if the number of halting equal to 5, we set $q_0 = 0.7(or0.6)$ to search intensively around the good solution and to accelerate convergence.

3.6. *Number of multi-populations and performance analysis*

parallel implementations aim to provide further diversity by using multi-populations and inter-population migration strategies. In the algorithm, all individuals are divided into some independent sub-colonies, called universes, each universe evolving independently uses the improved 3-opt algorithm, and information among the universes is exchanged by adopting emigration based on learning mechanism and adaptive quantum interaction simulating entanglement of quantum. The experimental results show that PQACO can obtain finer approximate result with less time by more sub-colonies. We conclude that PQACO tends to produce larger sets of search points adaptively and can improve solution quality more.

4. Experimental Study

In this section, The PQACO was implemented in C# and the tests were performed on a 400 MHz UltraSparc2 processor. We mainly examine the impact of different parameter settings on solution quality. In our computational experiment, TSP benchmark instances are Chn144.The average performance of each variant was calculated over 10 different runs, based on 200 iterations. We used m = 500 ants in all experiments. The pheromone evaporation ρ=0.2(see Eq. (2)).

The result is shown in Fig.5. The experimental results show that PQACO can obtain finer approximate result with less time by different parameter settings. The best result is 30356.77549.

The average performance of 10 different runs is shown in Fig.6, Fig.7 and Fig.8 by different parameter q_0 . The experimental results show that PQACO can obtain finer approximate result with less time by adaptive parameter. We conclude that PQACO tends to produce larger sets of search points adaptively and can improve solution quality more.

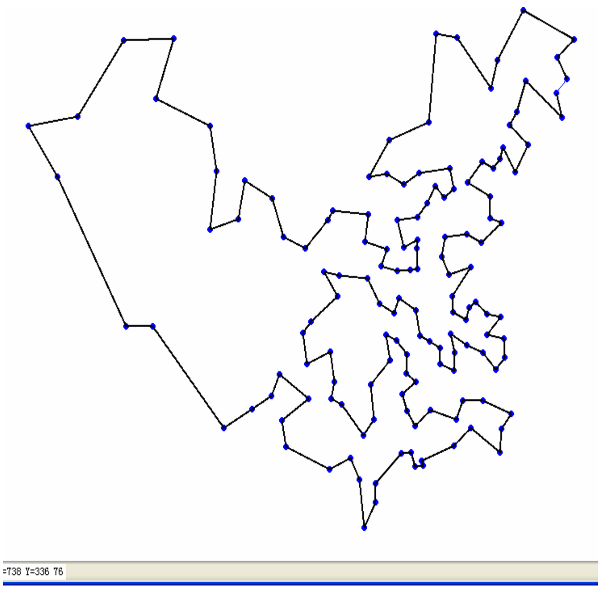

Figure 5.The best result of TSP benchmark instances Chn144

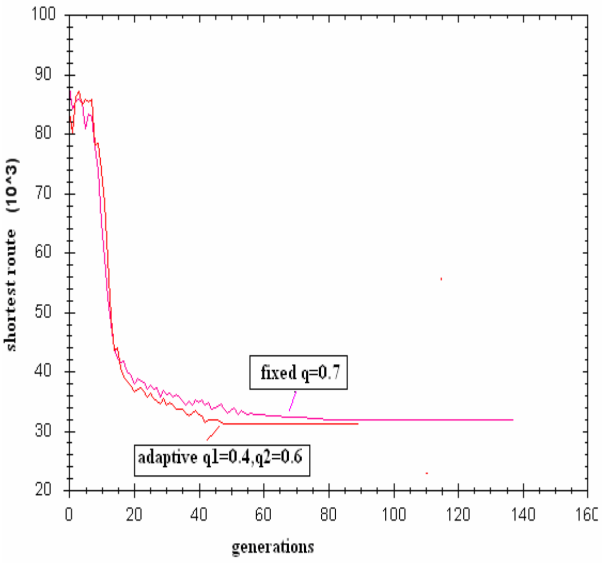

Figure6. The average result 1 of TSP benchmark instances Chn144 by different parameter q

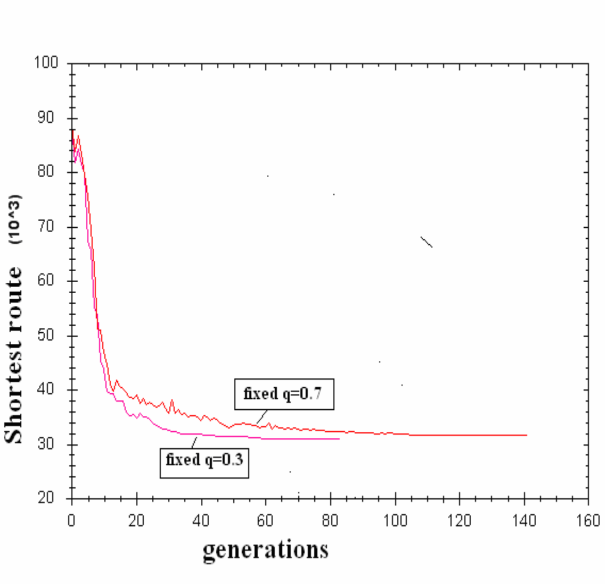

Figure7. The average result 2 of TSP benchmark instances Chn144 by different parameter *q*

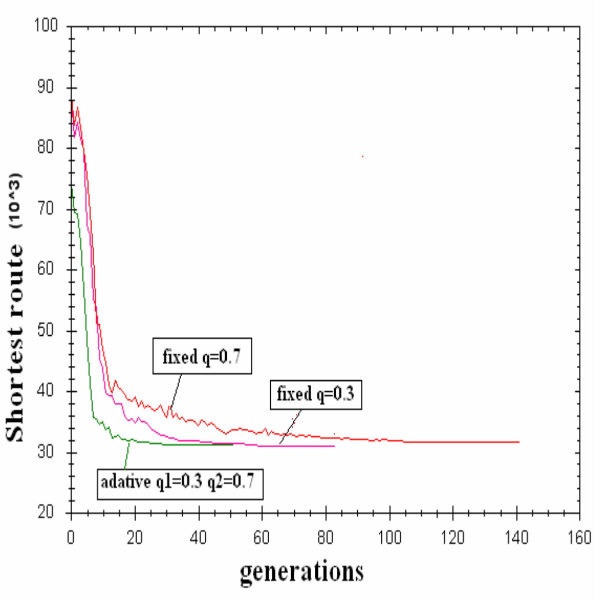

Figure8. The average result 3 of TSP benchmark instances Chn144 by different parameter q

The average performance of 10 different runs is shown in Fig.9 to Fig.15 by different number of sub-populations. The experimental results show that PQACO can obtain finer approximate result with less time by more subpopulations. We conclude that PQACO tends to produce larger sets of search points by more populations and can improve solution quality more by co-evolution.

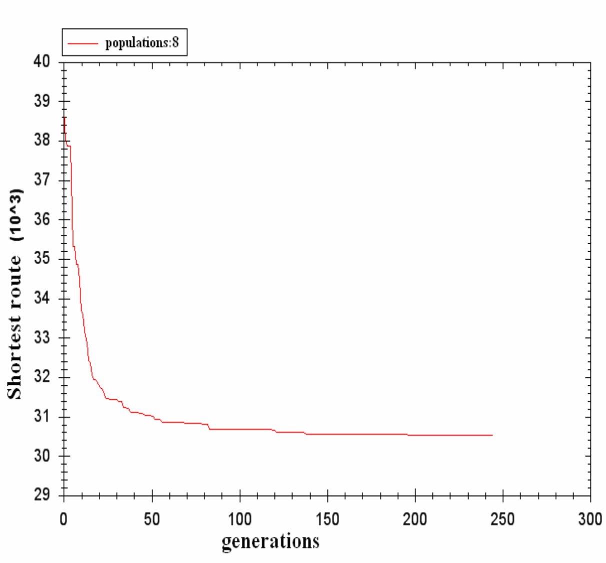

Figure9. The average result 1 of TSP benchmark instances Chn144 by different number of subpopulations

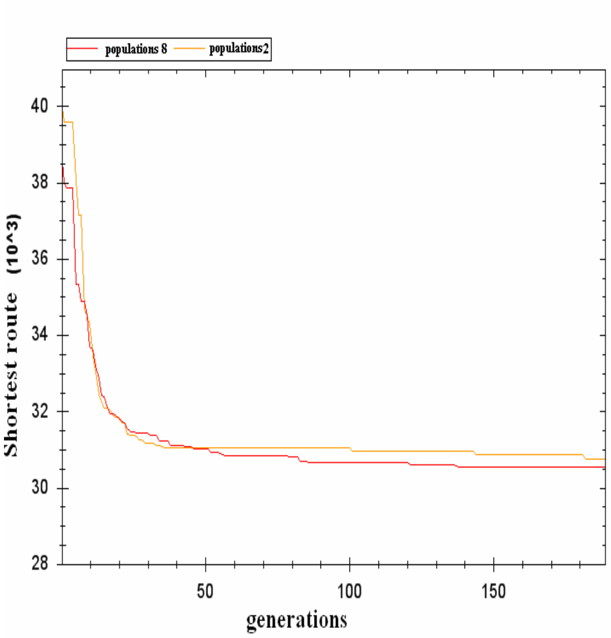

Figure10. The average result 2 of TSP benchmark instances Chn144 by different number of subpopulations

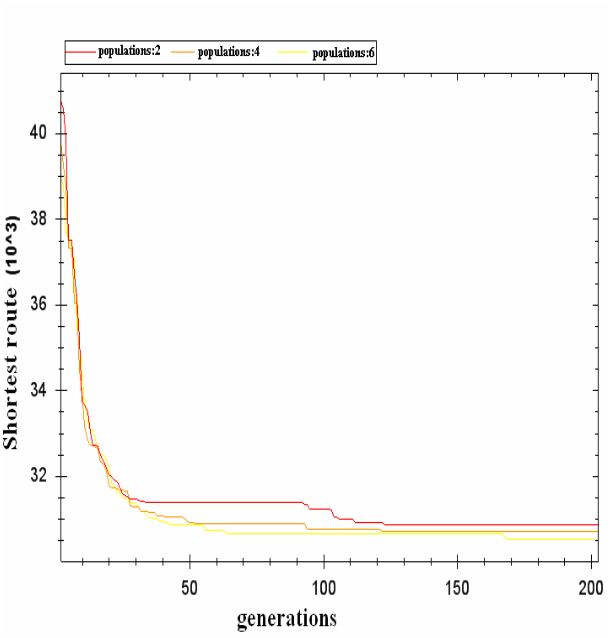

Figure11. The average result 3 of TSP benchmark instances Chn144 by different number of subpopulations

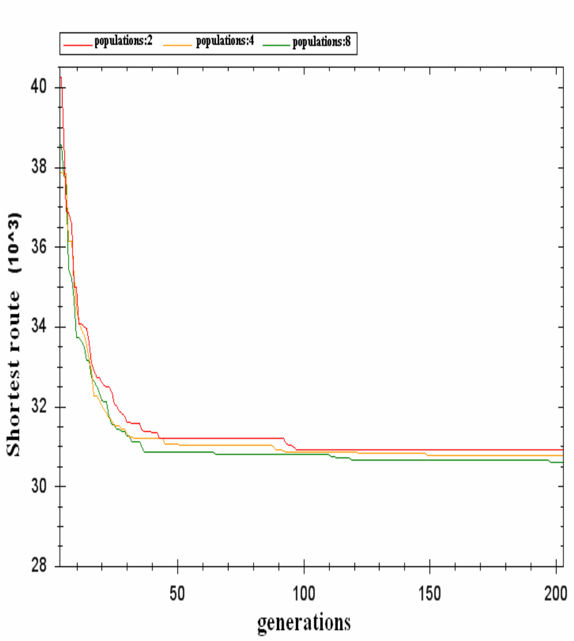

Figure12. The average result 4 of TSP benchmark instances Chn144 by different number of subpopulations

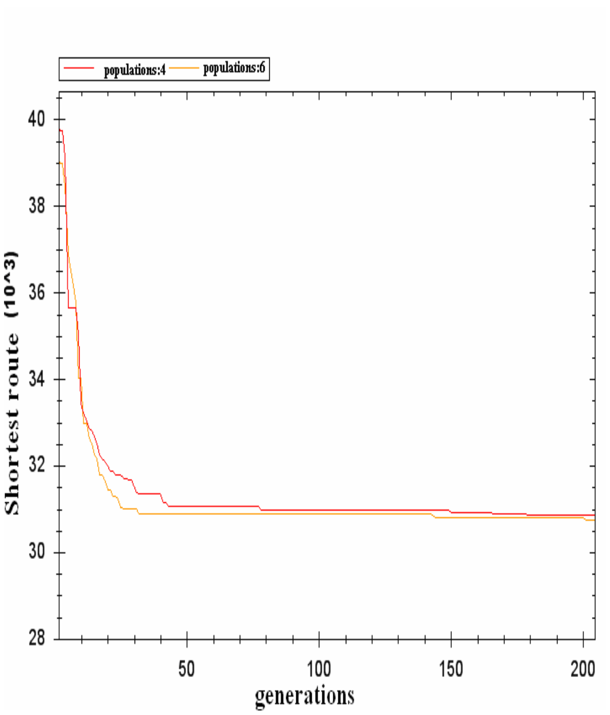

Figure13. The average result 5 of TSP benchmark instances Chn144 by different number of subpopulations

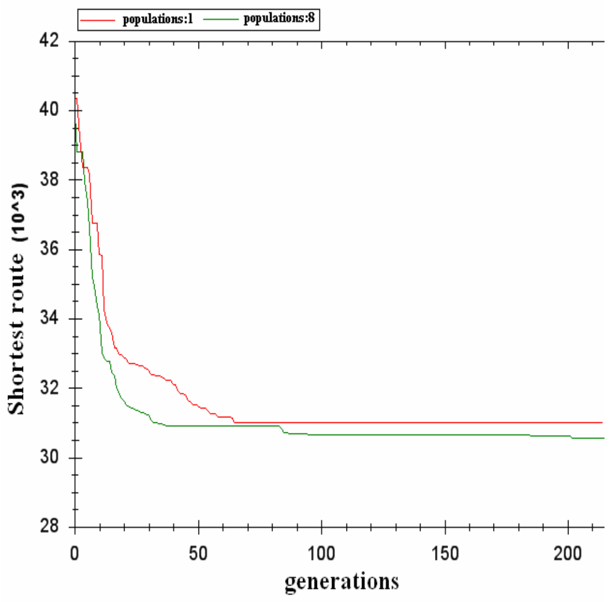

Figure14. The average result 6 of TSP benchmark instances Chn144 by different number of subpopulations

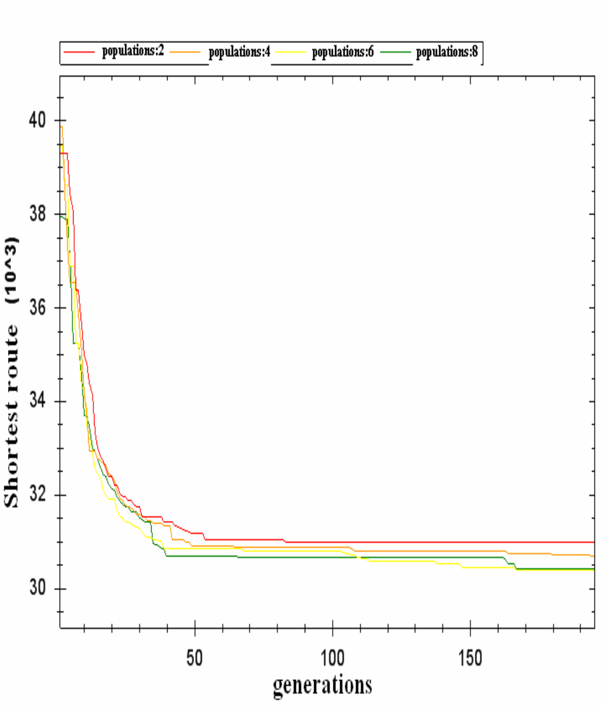

Figure15. The average result 7 of TSP benchmark instances Chn144 by different number of subpopulations

5. Concluding remarks and future work

In this study, a novel Quantum Dynamic Mechanism-based Parallel Ant Colony Optimization Algorithm for traveling salesman problem (PQACO) is proposed. The efficiency of Ant Colony Optimization algorithm is enhanced by using the Quantum computing operator and parallel Mechanism. We describe the Quantum Dynamic Mechanism and analysis the technology of improving performance; the efficiency of the approach has been illustrated by applying to TSP benchmark instances Chn144. The results show that integration of the quantum computing operator in the Ant Colony Optimization procedure can yield improvements in both the convergence rate and solution quality. The next work is to exploit quantum dynamic mechanism further to deal with the parameter setting of PQACO.

Acknowledgements

The authors gratefully acknowledge the support of National Natural Science Foundation of Shanghai (Grant No.09ZR1420800),National Natural Science Foundation of China(Grant No.61075115),Development Foundation of SUES (Grant No.2008XY18) and Doctor Foundation of SUES.

References

1. Narayanan and M. Moore, Genetic quantum algorithm and its application to combinatorial optimization problem, *in: Proc. IEEE International Conference on Evolutionary Computation (ICEC96)*, IEEE Press, Piscataway, 1996, pp.61-66.
2. Shuai, D., Shuai, Q. and Liu, Y at el, Particle Model to Optimize Enterprise Computing, *in International Federation for Information Processing, Volume 205, Research and Practical Issues of Enterprise Information Systems,* (Boston:Springer), 2006, pp.109-118.
3. X.M. You, S. Liu, and D.X. Shuai, Quantum Evolutionary Algorithm Based on Immune Theory, *In proc of WCICA06*, 2006.6,pp. 3410-3414.
4. K.H. Han, J.H. Kim, Quantum-Inspired Evolutionary Algorithms with a New Termination Criterion, H Gate, and Two-Phase Scheme, *IEEE Transactions on Evolutionary Computation*, **8**(2004),pp.156-169.
5. K.H. Han, J.H. Kim, Quantum-inspired evolutionary algorithm for a class of combinatorial optimization, *IEEE Transactions on Evolutionary Computation,* **6**(2002), pp. 580-593.
6. X.M. You, S. Liu, and D.X. Shuai, Quantum Evolutionary Algorithm Based on Immune Theory for Multi-Modal Function Optimization, *Journal of petrochemical universities,* 2007, **20**(3) pp.45-49.
7. Dorigo, M., Maniezzo, V. and Colorni, A., Positive feedback as a search strategy. *Tech. Report No.91-016*, Dipartimentodi Elettronica, Politecnico di Milano, Italy (1991).
8. Dorigo, M. and Blum, C., Ant colony optimization theory: A survey. *Theoretical Computer Science,* 2005, **344**(2-3), pp.243-278.
9. Dorigo, M., Gambardella and L.M., Ant colony system: A cooperative learning approach to the traveling salesman problem ,IEEE Trans. Evolutionary Computation,1997,**1**(1): pp.53-66.
10. M. G. Arenas, E. Parras-Gutiérrez and V. M. Rivas at el, parallelizing the Design of Radial Basis Function Neural Networks by Means of Evolutionary Meta-algorithms. *Lecture Notes in Computer Science*, 2009, Volume **5517**, pp.383-390.
11. Ting Hu, Simon Harding and Wolfgang Banzhaf, Variable population size and evolution acceleration: a case study with a parallel evolutionary algorithm, *Genetic Programming and Evolvable Machines*, 2010, Volume **11**, pp.205-225.

12. Liang Hu, Xilong Che and Xiaochun Cheng, Bandwidth Prediction based on Nu-Support Vector Regression and Parallel Hybrid Particle Swarm Optimization, *International Journal of Computational Intelligence Systems*. 2010, **3**(1). Pp.70-83.