Research on Traveling Salesman Problem Based on the Ant Colony Optimization Algorithm and Genetic Algorithm

Yu Chen* and Yanmin Jia

School of Civil Engineering, Northeast Forestry University, Harbin 150080, China

Abstract: In this paper, we prompt a new multi-dimensional algoithm to solve the traveling salesman problem based on the ant colony optimization algorithm and genetic algorithm. Ant Colony Optimization (ACO) is a heuristic algorithm which has been proven a successful technique and applied to a number of combinatorial optimization (CO) problems. The traveling salesman problem (TSP) is one of the most important combinatorial problems. ACO is taken as one of the high performance computing methods for TSP. It still has some drawbacks such as stagnation behavior, long computational time, and premature convergence problem of the basic ACO algorithm on TSP. Those problems will be more obvious when the considered problems size increases. The proposed system based on basic ACO algorithm with well distribution strategy and information entropy which is conducted on the configuration strategy for updating the heuristic parameter in ACO to improve the performance in solving TSP. Then, ACO for TSP has been improved by incorporating local optimization heuristic. Algorithms are tested on benchmark problems from TSPLIB and test results are presented. From our experiments, the proposed algorithm has better performance than ACO algorithm.

Keywords: Traveling Salesman Problem, Collabrative Algorithm, Ant Colony Optimization Algorithm, Genetic Algorithm.

1. INTRODUCTION

In recent years, many research works have been devoted to ant colony optimization (ACO) techniques in different areas. It is a relatively novel meta-heuristic technique and has been successfully used in many applications especially problems in combinatorial optimization. ACO algorithm models the behavior of real ant colonies in establishing the shortest path between food sources and nests. Ants can communicate with one another through chemicals called pheromones in their immediate environment. The ants release pheromone on the ground while walking from their nest to food and then go back to the nest. The ants move according to the amount of pheromones, the richer the pheromone trail on a path is, the more likely it would be followed by other ants. So a shorter path has a higher amount of pheromone in probability, ants will tend to choose a shorter path. Through this mechanism, ants will eventually find the shortest path. Artificial ants imitate the behavior of real ants, but can solve much more complicated problem than real ants can.

ACO has been widely applied to solving various combinatorial optimization problems such as Traveling Salesman Problem (TSP), Job-shop Scheduling Problem (JSP) [1, 2], Vehicle Routing Problem (VRP), Quadratic Assignment Problem (QAP), etc. Although ACO has a powerful capacity to find out solutions to combinational optimization problems, it has the problems of stagnation and premature convergence and the convergence speed of ACO is very slow. Those problems will be more obvious when the problem size in creases [3]. Therefore, several extensions and improvements versions of the original ACO algorithm were introduced over the years. Various adaptations: dynamic control of solution construction [4], emergence of local search [5, 6], a strategy is to partition artificial ants into two groups: scout ants and common ants [7] and new pheromone updating strategies [8], using candidate lists strategies are studied to improve the quality of the final solution and lead to speedup of the algorithm. All these studies have contributed to the improvement of the ACO to some extents, but they have little obvious effect on increasing the convergence speed and obtaining the global optimal solution. In the proposed system, the main modifications introduced by ACO are the following. First, to avoid search stagnation and ACO is more effective if ants are initially placed on different cities. Second, information entropy is introduced which is adjust the algorithm's parameters. Additionally, the best performing ACO algorithms for the TSP improve the solutions generated by the ants using local search algorithms. The experiment results show that the algorithm proposed in this study can substantially increase the convergence speed of the ACO.

In this paper, an improved ant colony optimization algorithm is developed for solving TSP. This algorithm is used to produce near-optimal solutions to the TSP. Evolution Algorithms are widely used optimization algorithm based on Darwinism. In Darwinism, evolution is based on survival of the fittest. Creatures do not have the ability of deciding their mutation directions and choosing advantageous gene to their offspring. But based on the researches in biology, this stochastic evolution theory could not explain some problems in creature's adaptability.

*Address correspondence to this author at the School of Civil Engineering, Northeast Forestry University, Harbin 150080, China; Tel: +860000000000; E-mail: 34713843@qq.com

The modern biology research shows that the evolution process is not completely stochastic. Such as stress-induced mutation mechanisms were proposed. Specifically, when creatures are maladapted to their environment, that is, when they are stressed, stress-induced mutation mechanisms produce mutations. This theory could be introduced into evolution algorithms in solving some optimization problems. In this paper, an approximate algorithm combining P systems and active evolution algorithm (AEAPS) is proposed in order to solve traveling salesman problems (TSPs} in the special case of complete graphs with Euclidean distance. It follows the nested membrane structure adopted by Nishida, and adopts genetic algorithm (GA), tab search and active evolution algorithm (AEA) as the sub algorithms. The experiments results are compared with Nishida's algorithms and some others.

2. THE BACKGROUND AND FRAME WORK OF GENETIC ALGORITHM

Genetic algorithm is the late 1960s and early 1970s, by John Holland and his colleagues at the University of Michgan American student research a method of forming the study. In this method, mechanism simulation of biological evolution Model to construct artificial system, has been widely used in recent years. From Traditional BP neural network algorithm slow convergence and easy to fall into Local minima shortcomings, this paper based on genetic algorithm BP Network mechanisms to improve the convergence speed of the network, and then using the improved BP Neural networks for evaluation of the level of the university library information.

In this paper, three-layer BP network, shown in Fig. (**1**). BP network three Layer nodes are represented, m input nodes x_i (each two indicators), in a hidden layer node h_j, 1 output node y. Network power input nodes and hidden layer nodes Value w_{ij}, network nodes and the output power hidden layer nodes is t_j.

Each index can score from reviewer's subjective scoring method after obtaining. The data to be using equation (1) is normalized.

$$\overline{x_i} = \frac{x_i - b_i}{a_i - b_i} \tag{1}$$

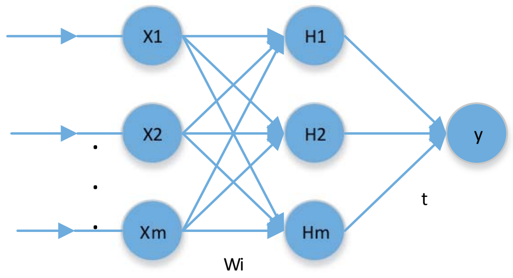


Fig. (1). Three BP neural network.

Where x_i and $\overline{x_i}$ respectively represent the i-th index and the actual value standard value; a_i, b_i are the maximum, minimum, the i-th index. Known evaluation indexes m, n hidden layer nodes depending Problems and experimental data to determine, you can also experience the value of the formula (2) the decision.

$$n = \log_2 m \tag{2}$$

Hidden node output is calculated as follows:

$$h_j = f\left(\sum_{i=1}^{m} w_{ij} x_i - \theta_j\right) \tag{3}$$

Where θ_j is the threshold value for hidden node.

The output of the output node is calculated as follows:

$$f\left(\sum_{i=1}^{m} w_{ij} x_i - \theta_j\right) = f\left(f\left(\theta_j\right)\right) \tag{4}$$

Where in θ is an output node threshold.

Equation (3) and Equation (4) in the transfer function is generally expressed as (0,1) interval of S-type function:

$$f\left(x\right) = \frac{1}{1+e^{-x}} \tag{5}$$

Using genetic algorithm to train the neural network can be used in binary code, code String value from the hidden layer to the input layer connection weights w_{ij}, hidden layer and output layer connection Weights t_j, hidden layer threshold θ_j, the output layer threshold θ. So connected together to form into a long string (the string corresponding to each position of a group of network weights and threshold Value), constitutes an individual. It can generate an initial population of N individuals.

Guiding the evolutionary process toward the region of space may contain the best individual conduct. Suitable Select the function should have a great impact on the training results. This paper fitness Function uses a calculation error of the neural network (see equation 6), and the error. The greater the difference is, the smaller the fitness is. Always cut along the neural network output error fewer search direction, the convergence target is the minimum output error.

$$E = \frac{1}{N}\sum_{s=1}^{N}\sqrt{\left|\left(\overset{s}{y}\right)^2 - \left(\overset{\sim s}{y}\right)^2\right|} \tag{6}$$

Where $\overset{s}{y}, \overset{\sim s}{y}$, respectively, for the desired output and the actual output, N for training.

Training samples, then the fitness function is:

$$f\left(X_s\right) = \frac{1}{E} \tag{7}$$

Which, Xs represents one individual.

The basic genetic algorithm by selection, crossover and mutation operators constitute three. Select operation is based assessment in groups of individual fitness value Conduct random roulette method adopted in this algorithm on the basis of estimates Selection. Each chromosome number of offspring produced. According to individual fitness the order value is converted to the corresponding selection probability, a large fitness value corresponding High probability of selection, small fitness value corresponding to a low probability of selection [2], each chromosome Selection probability p_s. The formula is:

$$p_s = c\left(1-c\right)^{s-1} \tag{8}$$

Where the parameter c is the probability of an individual that is selected the first sorting. In the selection wheel when using binary search method to effectively reduce A small number of comparisons to ensure that the corresponding roulette found (N in O (1og N) For population size, the number of training samples).

When using consistent cross crossover method (ie, word by setting shield The entire body of inherited genes determine the two parent individuals in which the individual The corresponding gene, wherein the shield is a randomly generated word and individual encoding result 0/1 sequence identical configurations).

When variation in a population of randomly selected individuals, mutation operator Loci child adoption, the basic steps are as follows: start with groups of individuals compiled Code string randomly selected one or more loci, and then follow the variation The probability of each generation after the completion of genetic manipulation, for a new generation of individual error and evaluate the fitness value, to identify the best individual, the optimal individual two-step gradient descent local search, and put out a search of the directly into the body of the new generation of the population, if the learning error achieve accuracy or up to specify the genetic algebra calculation is determinate. So after more than genetic luck count, you can get BP neural network error minimum set of complete initial Weights and thresholds.

Genetic networks by BP's initial iterations to get the right value and threshold for normal network training decoded offspring into BP neural network until the end of the study. This method using genetic algorithms to search after the near global optimum, then using BP algorithm local search such BP algorithm can quickly and accurately converge near the solution point, final get the problem global minimum. In this way, both to avoid the BP neural network into local minimum value and BP neural network can quickly close convergence, greatly improving the accuracy and speed of the BP neural network.

The BP neural network is widely used mainly for the following advantages: giant parallel distribution makes the processing speed greatly; the distribution of knowledge information is stored in network connection and threshold that has strong fault-tolerance. Although BP network model has many advantages in all aspects, but also has some deficiencies: from the view of mathematics, it is a nonlinear optimization problem and inevitably there exists local minimum points; study algorithm is very slow; the selection of network hidden nodes with great blindness, and there are no empirical theoretically instructions and etc. Therefore, these disadvantages greatly limit the application and promotion of the BP neural network. The paper establishes the highway project construction risk prediction model, mainly using genetic algorithms and BP network combination to improve the performance of BP network, on the basis of series of improvement such as the sample data pretreatment, optimize network hidden nodes, introducing momentum, adaptive adjustments network parameters.

Genetic algorithm is a search algorithm based on natural selection and genetic mechanisms for groups, and simulates the course of reproduction, and the interbreeding in the process of a natural and genetic selection. Using genetic algorithm, every possible solution is coded as "a chromosome", namely the individual, a number of individuals form groups (all possible solutions). Genetic algorithm is the process of evolution generational group which is the community of a feasible solution.

The genetic algorithm begins, always randomly generated some individuals (i.e. initial weights and threshold), according to the predetermined target function (i.e. the error function) for each individual evaluation, then gives a fitness value. Based on the fitness value, choose the individual to copy the next generation. Choosing reflects the operation principle of "survival of the fittest", "good" individual was used to copy, and the "bad" individual will be eliminated. Then the individual combined the intersection and operator variations to create a new generation. This is a new group of individual which inherited some excellent characters of the generation, thus the performance is superior to the generation, so gradually toward more optimal threshold value and the direction of evolution. When the evolution reached accuracy requirement, and can be asked during a series of network weights and threshold.

The GA search always pervades all of the space, which is good at the global search; While BP neural network for local search is more effective. Therefore, both of them combine and complement with each other. First use genetic algorithms to optimize BP neural network, set a good search spaces, then use BP algorithm to search out the optimal solution in this little space.

3. THE BASIC ACO ALGORITHM

In this section we introduce the basic ACO algorithm. We decided to use the well-known traveling salesman problem as benchmark, in order to make the comparison with other heuristic approaches easier. Given a set of ri towns, the TSP can be stated as the problem of finding a minimal length closed tour that visits each town once. We call d_{ij} the length of the path between towns i and j. In the case of Euclidean TSP, d_{ij} is the Euclidean distance between i and j (i.e., $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$). An instance of the TSP is given by a graph(N,E), where N is the set of towns and E is the set of edges between towns (a fully connected graph in the Euclidean TSP). Let $b_i(t)$ (i=1,2,...,n) be the number of ants in town at time t and let $m=\sum_{i=1}^{n} b_i(t)$ be the total number of ants. Each ant is a simple agent with the following characteristics: it chooses the town to go to with a probability that is a function of the town distance and of the amount of trail present on the connecting edge to force the ant to make legal tours, transitions to already visited towns are disallowed until a tour is completed (this is controlled by a labu list). When it completes a tour, it lays a substance called trail on each edge (i,j) visited.

Let $\tau_{ij}(t)$ be the intensity of trail on edge (i,j) at time t. Each ant at time t chooses the next town, where it will be at time t+1. Therefore, if we call an iteration of the ACO algorithm the m moves carried out by the m ants in the interval (t,t+1), then every n iterations of the algorithm (which we call a cycle) each ant has completed a tour. At this point the trail intensity is updated according to the following formula:

$$\tau_{ij}(t+n)=\rho\tau_{ij}(t)+\Delta\tau_{ij} \tag{9}$$

where ρ evaporation is a coefficient such that (1-ρ) represents the of trail between time t and t+n

$$\Delta\tau_{ij}=\sum_{k=1}^{m}\Delta\tau_{ij}^{k} \tag{10}$$

Where $\Delta\tau_{ij}^{k}$ is the quantity per unit of length of trail substance (pheromone in real ants) laid on edge (i,j) by the k-th ant between time t and t+n . It is given by

$$\Delta\tau_{ij}^{k}=\begin{cases}\dfrac{Q}{L_K} & \begin{array}{l}\text{if k – th ant uses}\\ \text{edge (i, j) in its tour}\\ \text{(between time t and t+n)}\end{array}\\ 0 & \mathit{otherwise}\end{cases} \tag{11}$$

where Q is a constant and L_k is the tour length of the k-th ant. The coefficient ρ must be set to a value ρ<1 to avoid unlimited accumulation of trail. In our experiments, we set the intensity of trail at time 0, $\tau_{ij}(0)$, to a small positive constant c.

In order to satisfy the constraint that an ant visits all the n different towns, we associate with each ant a data structure called the tabu list, that saves the towns already visited up to time t and forbids the ant to visit them again before n iterations (a tour) have been completed. When a tour is completed, the lobo list is used to compute the ant's current solution (i.e., the distance of the path followed by the ant).

The tabu list is then emptied and the ant is free again to choose. We define tabu_k the dynamically growing vector, which contains the lobo list of the kth ant, tabu_k the set obtained from the elements of tabu_k and tabu_k(s) the s-th element of the list (i.e., the s-th town visited by the k-th ant in the current tour).

We call visibililty η_{ij} the quantity η_{ij}. This quantity is not modified during the run of the ACO algorithm, as opposed to the trail, which instead changes according to the previous formula (1). We define the transition probability from town i to town j for the k-th ant as

$$p_{ij}^{k}(t)=\begin{cases}\dfrac{\tau_{ij}^{\alpha}(t)\cdot\eta_{ij}^{\beta}}{\sum\limits_{s\in allowed_k}\tau_{is}^{\alpha}(t)\cdot\eta_{is}^{\beta}} & j\in allowed_k\\ 0 & otherwise\end{cases} \tag{12}$$

where

$$allowed_k=\left\{N-tabu_k\right\} \tag{13}$$

And where α and β are parameters that control the relative importance of trail versus visibility. Therefore the transition probability is a trade-off between visibility (which says that close towns should be chosen with high probability, thus implementing a greedy constructive heuristic) and trail intensity at time t (that says that if on edge (i, j) there has been a lot of traffic then it is highly desirable, thus implementing the autocatalytic process).

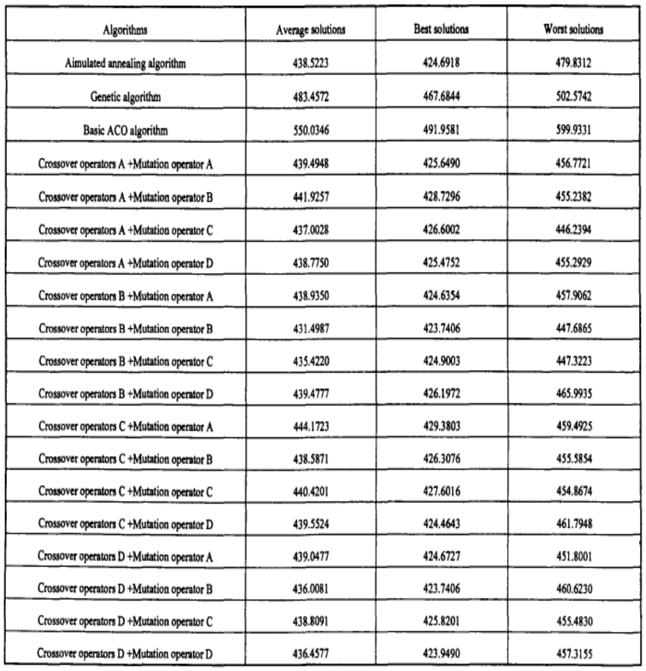
In traditional ACO Algorithm, the initialization of the pheromone matrix is equal. Ants need iterate many numbers to find the best tour. We can generate a large amount of tours (*e. g.* 100 tours), and then we choose some better tours (*e. g.* 30 tours). At last ants lay trail only on these better tours. These trails affect following ants.

When ant completes a tour, it always lays called trail on each edge visited. If the tour is worse, ant also lay the trail on each edge. These trails disturb the following ants, so the ACO algorithm's convergent speed is very slow. We can calculate the length of tour firstly and then we compare with the given value. If the length of tour is less than the given value, we update trail values. Otherwise don't update the trail values.

The advantage of PSO algorithm method is that it use self information, individual best information and global best information. We can learn from genetic algorithm, and were arrange formula (12). The $w\times v_{id}$ can be regarded as mutation operator, and $c_1\times rand_1\times(p_{id}-x_{id})+c_2\times rand_2\times(p_{gd}-x_{id})$ can be regarded as crossover operator. But the crossover operator happened between the individual with local optimum and global optimum.

4. EXPERIMENTAL ANALYSIS

This section compares the results of simulated annealing algorithm, genetic algorithm, ACO algorithm and hybrid algorithms on traveling salesman problem of 30 cities. The parameters of simulated annealing algorithm are set as follows: the initial temperature T=100000, the final temperature T_0=1, and annealing velocity α = 0.99 .The parameters of the genetic algorithm optimization toolbox (GAOT) used to solving TSP are set as follows: the population N=30, the cross probability P_c= 0.2 and the mutation probability P_m= 0.5 The parameters of the hybrid algorithms are set as follows: α =1.5, m=30, β =2 ,and ρ =0.9. 20 rounds of computer simulation are conducted for each algorithm, and the results are shown in Fig. (**2**). The optimal tour of 30 cities by hybrid algorithm is shown in Fig. (**3**). All the 16 hybrid algorithms are proved effective. Especially the hybrid algorithm with across strategy B and mutation strategy B is a simple and effective better algorithm than others. The crossover operators of particle swarm optimization are different from those of genetic algorithm. In hybrid optimization, the crossover operator happened between the individual with local optimum and global optimum, so the capability of offspring got to be improved.



Algorithms	Average solutions	Best solutions	Worst solutions
Aimulated annealing algorithm	438.5223	424.6918	479.8312
Genetic algorithm	483.4572	467.6844	502.5742
Basic ACO algorithm	550.0346	491.9581	599.9331
Crossover operators A +Mutation operator A	439.4948	425.6490	456.7721
Crossover operators A +Mutation operator B	441.9257	428.7296	455.2382
Crossover operators A +Mutation operator C	437.0028	426.6002	446.2394
Crossover operators A +Mutation operator D	438.7750	425.4752	455.2929
Crossover operators B +Mutation operator A	438.9350	424.6354	457.9062
Crossover operators B +Mutation operator B	431.4987	423.7406	447.6865
Crossover operators B +Mutation operator C	435.4220	424.9003	447.3223
Crossover operators B +Mutation operator D	439.4777	426.1972	465.9935
Crossover operators C +Mutation operator A	444.1723	429.3803	459.4925
Crossover operators C +Mutation operator B	438.5871	426.3076	455.5854
Crossover operators C +Mutation operator C	440.4201	427.6016	454.8674
Crossover operators C +Mutation operator D	439.5524	424.4643	461.7948
Crossover operators D +Mutation operator A	439.0477	424.6727	451.8001
Crossover operators D +Mutation operator B	436.0081	423.7406	460.6230
Crossover operators D +Mutation operator C	438.8091	425.8201	455.4830
Crossover operators D +Mutation operator D	436.4577	423.9490	457.3155

Fig. (2). Testing result of algorithms.

CONCLUSION

In this paper, we presented a novel algorithm of genetic ant colony optimization (GACO) for traveling salesman problem. It keeps the advantages of ant colony optimization and GAs. From our simulation for those test problems, the proposed algorithm indeed can find the best solutions or optimal solutions. In other words, the proposed algorithm seems to have admirable performance. Experiments for benchmark problems show the hybrid algorithm better than other algorithms. The following problems need to be considered.

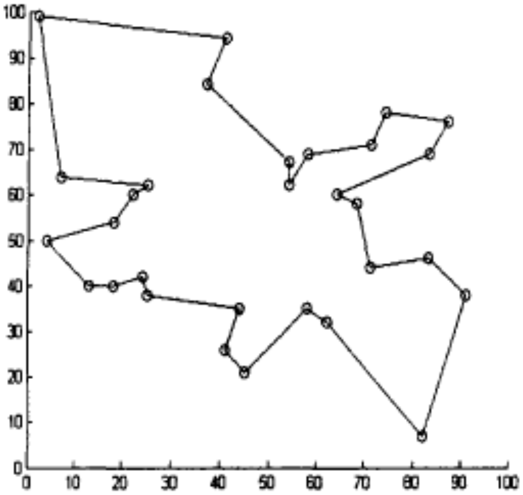


Fig. (3). The optimal tour of 30 cities by hybrid algorithm.

1. The parameters and their effect on the performance of the optimization should be studied in more detail.

2. How to explore hybrid algorithm application to continuous space problem should be investigated.

3. The hybrid algorithm's convergent speed, or the efficiency, should be worth further investigating.

4. How to evaluate the quality of hybrid algorithm and other algorithms is still a problem.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

This work is supported by the Research project of science and technology of Inner Mongolia communication and transportation bureau.

REFERENCES

[1] Q. Zhao, G. Zhao, and J. Y. Wang, “University education level of information and comprehensive measure cooperative research”, *Intelligence Magazine,* no. 6, pp. 101-102, 2004.

[2] J. Tai, W. Meleis, and J. Zhang, “Adaptive resource allocation for cloud computing environments under bursty workloads”, pp. 978-987, Northeastern University, Boston, USA, 2013.

[3] J. M. Tirado, D. Higuero, and F. Isaila, “Predictive data grouping and placement ivor cloud-based elastic server infrastructures”, In: *11th IEEE/ACM International Symposium on Cluster, Cloud and grid Computing. IEEE*, 2011, pp. 281-294.

[4] S. I. U. Labs, “Private cloud computing with open Nebula”, In: *Software System: Private Cloud Computing with Open Nebula,* 2013, pp. 21-30.

[5] Z.Y. Guo, Z.Y. Chen, L.Q. Li, B.P. Song, and Y. Lu, “Artificial neural network and its application in regime prediction of groundwater quality,” *Journal of East China Normal University (Natural Sciences),* no. 1, pp. 84-89, 2001. (in Chinese).

[6] R. Z. Li, “Advance and trend analysis of theoretical methodology for water quality forecast,” *Journal of Hefei University of Technology*, vol. 29, no.1, pp. 26-30, 2006.

[7] H.F. Mo, A.Y. Gu, X.Z. Zhang, and J.C. Zhang, “Research on a method of BP neural network in water quality evaluation,” *Control Engineering of China*, vol. 11, pp. 9-10, 2004.

[8] Z.G. Niu, H.W. Zhang, and H.B. Liu, “Application of neural network to prediction of coastal water quality,” *Journal of Tianjin Polytechnic University,* vol. 25, no .2, pp. 89-92, 2006.

Received: February 2, 2015 Revised: April 29, 2015 Accepted: May 18, 2015