Available online at www.sciencedirect.com

SciVerse ScienceDirect

Systems Engineering Procedia 4 (2012) 226 – 235

Procedia
Systems Engineering

The 2nd International Conference on Complexity Science & Information Engineering

Comparison of several intelligent algorithms for solving TSP problem in industrial engineering

Wang Hui[a]

North China Electric Power University No.2 Beinong Rd., Huilongguan, Beijing, 102206, P.R.China

Abstract

The paper presents three intelligent algorithms, namely, basic genetic algorithm, Hopfield neural network and basic ant colony algorithm to solve the TSP problem. Then different algorithms are compared in the perspectives of time complexity, space complexity, the advantages and disadvantages of the calculation results, and difficulty level of realization. We use the application of paired comparison matrix to make comprehensive evaluation, and then give the value of comprehensive evaluation in engineering.

Keywords: basic genetic algorithm, Hopfield neural network, ant colony optimization, paired comparison matrix, TSP

1. Introduction

The travelling salesman problem (TSP) is a problem in combinatorial optimization studied in operations research and theoretical computer science and in engineering. Given a list of cities and their pair wise distances, the task is to find a shortest possible tour that visits each city exactly once.This is an NP-hard problem, when a large number of nodes of G, if the use exhaustive search, the time complexity is $O(n!)$, if use the search of dynamic programming, the time complexity is $O(n^2 2^2)$, combinatorial explosion will occur in the both search methods. Therefore, the majority of domestic and foreign scholars began to study intelligence algorithms for TSP, since the basic genetic algorithm appears, they began to examine the use of genetic algorithm on solve TSP problems until present and proposed many improvements. Reference [1] presents a genetic algorithm based on common path, reference [2] proposed a new genetic algorithm through using multiple-search method. All these improved genetic algorithms are promising approach for TSP problem. Hopfield network was proposed in 1970s, and in 1985, Hopfield proposed to use CHNN for solving TSP problems, but Hopfield network prone to ineffective solutions and local solutions, so many scholars have been studying how to improve the algorithm, reference [3] analyzed the effectiveness of solving TSP with Hopfield, reference [4] through optimizing the Hopfield network and path of the initial steps to improve the Hopfield network to solve TSP and received good results. Ant colony algorithm which is effectiveness proposed a new computational intelligence algorithm for solving TSP problems recently. Because of its use of pheromone heuristic function, can greatly reduce the search

* Corresponding author. Tel.+086-010-51963765; fax: +086-010-51963765.
E-mail address: wanghuibd@163.com.

2211-3819 © 2011 Published by Elsevier Ltd. Selection and peer-review under responsibility of Desheng Dash Wu.
doi:10.1016/j.sepro.2011.11.070

space, so compared to other algorithms it have better time performance, but precisely because of this, ant colony algorithm is easy to fall into local optimal solution, reference [5] by dynamically adjusting the pheromone volatility to present a dynamic ant colony algorithm and get a more satisfactory results. All of these three algorithms for solving TSP problems have advantages and disadvantages. In this paper, these three kinds of algorithms are compared in time complexity, space complexity, the advantages and disadvantages of the calculation results, and difficulty level of realization, etc. We apply paired comparison matrix to make comprehensive evaluation, and then give the value of comprehensive evaluation.

2. The basic steps of three algorithms

2.1. The application of genetic algorithm for TSP

This paper uses a common framework of genetic algorithm to solve TSP. this section gives the general framework of genetic algorithm, then given the steps of genetic operators' algorithm.

Common framework for basic genetic algorithm

STEP0 Q=generateinitialpopulation( )； // Initialize population Q

STEP1 F=calculateobjectfitness(Q)； // calculated fitness(F) of population Q

STEP2 FOR i=1 to T BEGIN // T is the iteration step

STEP3 selectoperator(Q,F)； // Selection operator

STEP4 crossoveroperator(Q,pc)； // crossover operator

STEP5 mutationoperator(Q,F,pm)； // mutation operator

STEP6 F=calculateobjectfitness(Q)； // calculated fitness(F) of population Q

STEP7 END

2.1.1. Encoding and decoding

This paper is using the decimal coding to encoding for the path. For example, a chromosome 193,264,785 that represent a path, the path from the starting city 1, followed city 9,3,2,6,4,7,8,5 and finish return to the city 1.

2.1.2. Fitness function

Fitness function is calculated by the reciprocal of the path distance, that is, the longer the path the smaller the fitness, and vice versa. Formula is: $f = \frac{1}{S}$, while $S = \sum_{i}^{n} d_{i,i+1}$, It represents path length.

2.1.3. Selection operator

We use classic roulette wheel method to select the operator. First we calculate the fitness of each individual. Then calculate the probability of individual to be selected and use roulette method to choice the next generation of individual.

2.1.4. Crossover operator

(1) Select two individuals which were recorded as X and Y from the population.

(2) Select two crossover points are denoted i and j randomly, generally $i < j$.

(3) We set a new individual T, make T = Y, then remove the cities which belong to X_i to X_j .The first bit to the $i-1$ of Z_m is same as the first bit to the $i-1$ of T_m . The i bit to the j of Z_m is same as the i bit to the j bit of T_m . then send the i bit and behind of the i bit of T_m to Z.then get the individual Z.

(4)Similarly, we can get another individual W. For example, let X = (1,2,3,4,5,6,7,8), Y = (3,4,5,1,7,8,2,6), i = 3, j = 5 , then Z = (1,7,3,4,5,6,2,8), W = (2,3,1,7,8,4,5,6)

2.1.5. Mutation operator

In this paper, we use a heuristic mutation operator[6]. Heuristic mutation operator use neighborhood heuristic techniques to improve the efficiency of future generations. We selected λgenes Randomly. Product neighborhood by transposition of all selected genes, then evaluate all neighbor points, select the best neighborhood to be offspring .For example, P: (264735891), selected three positions 2,6,8 by random, we can obtain five different individuals by change their positions: A1: (264739851), A2: (254736891), A3: (254739861), A4: (229473586 1), A5: (294736851) then choose the best variation to be offspring.

2.2 the appellation of Hopfield network for TSP

The core of Hopfield Network for TSP is to determine the network energy function and derive the equation of state of the network. We are running the network to reach a steady state, the steady-state is a solution for the problem.

2.2.1. The energy function of Hopfield network for TSP

reference[4] proposed a simplified energy function.

$$E=\frac{A}{2}\sum_i\left(\sum_x v_{xi}-1\right)^2+\frac{B}{2}\sum_i\left(\sum_x v_{xi}-1\right)^2+D\int_0^{v_{xi}}\sum_y d_{xy}v_{y,i+1}dv_{xi} \tag{1}$$

The state equation is:

$$\frac{du_{xi}}{dt}=-A\left(\sum_i v_{xi}-1\right)-B\left(\sum_Y v_{yi}-1\right)-D\sum_y d_{xy}v_{y,i+1} \tag{2}$$

$$v_{xi}=\frac{1}{2}\left[1+\tanh\left(\frac{u_{xi}}{\lambda}\right)\right] \tag{3}$$

2.2.2. The steps of Hopfield Network for TSP

SETP0 FOR t=1 to T // T is the number of iterations

STEP1 v_0=InitV()； //Initialize v0

STEP2 $u_0=\lambda \arctan h(2\times v_0-1)$;

STEP3 v=Hopfield(A,B,D,N,F, v_0, u_0, T1, d_t, λ)

STEP4 FOREACH v_i IN v

STEP5 IF v_i >=0.99 THEN v_i =1

STEP6 IF v_i <=0.01 THEN v_i =0

STEP7 END

STEP8 IF v is not valid THEN 'Ineffective solution'

STEP9 END

Which, T1 is the number of inner loop iterations of Hopfield network, dt is the length of iteration step, the value of initial vis from reference [4]. Hopfield network function uses the state equation which given in 2.2.1 to obtain the solution.

2.3 The application of ant colony algorithm for TSP

Initial ant colony algorithm is graph-based algorithms, is proposed by Gutjahr WJ in 2000. The steps of algorithm are as follows:

STEP0: The TSP for n cities. N={1,2,…,n}， A={(i,j), i,j∈ N}， The matrix for distance between cities (dij)n×n, We set value of each arc(i,j) is τij(0)=1/|A in TSP graph. We suppose m ants are working, all the ants are starting from the same city i0. The best solution current is w = (1,2, ..., n).

STEP1 (outer loop) if accordance with stopping rule of algorithm, stop the calculation and output the best solution which we calculate. Otherwise, the ant s starts from i0.L (s) represents the collection of cities which ant walked, and at first L (s) is the empty.

STEP2 (inner loop) We calculated according to the order 1≤s≤m of ants. When the ants in the city i, if $L(s)=N$ or $\{l \mid (i,l) \in A, l \notin L(s)\} = \Phi$,complete the calculation of s-ants. Otherwise, if $L(s) \neq N$ and $T = \{l \mid (i,l) \in A, l \notin L(s)\}$, $-\{i_0 \neq \Phi\}$,it will according with the probability $p_{ij} = \frac{\tau_{ij}(k-1)}{\sum_{l \in T} \tau_{ij}(k-1)}, j \in T$ to reach j , $L(s) = L(s) \cup \{j\}, i = j$ .if $L(s) \neq N$ and $T = \{l \mid (i,l) \in A, l \notin L(s)\} = \Phi, -\{i_0 \neq \Phi\}$,it will reach i_0 , $L(s) = L(s) \cup \{j\}, i = j$;then repeat STEP 2.

STEP3 For 1≤s≤m, if $L(s) = N$, we according to the order of city to calculation in the length of path. If $L(s) \neq N$, the length of path will set to an infinite value (not up).then compare in the path length of m ants, denoted by the shortest path of ant is t If $f(L(t)) < f(L(W))$,we will use the following formula for path W to strengthen the pheromone and reduce the pheromone on the other path, the function is :

$$\begin{pmatrix} \tau_{ij}(k) = (1-\rho_{k-1})\tau_{ij}(k-1) + \dfrac{\rho_{k-1}}{|W|} \cdots\cdots\cdots (i,j) \in W \\ \tau_{ij}(k) = (1-\rho_{k-1})\tau_{ij}(k-1) \cdots\cdots\cdots\cdots\cdots\cdots (i,j) \in W \end{pmatrix} \quad (4)$$

We will get new $\tau_{ij}(k), k = k+1$, then repeat steps STEP 1.

3. Comparison of three algorithms

3.1. Comparison of Time complexity

The time complexities of selection operator, crossover operator and mutation operator in genetic algorithm are $O(n_0 n), O(n_0 P_c n^2), O(n_0 P_m n^2)$. While n_0 is the initial size of population. For under normal circumstances $p_c > p_m$, so $O(n_0 P_c n^2) > O(n_0 P_m n^2) > O(n_0 n)$. T is the number of outer iterations, so the time complexity of outer loop is $O(Tn_0 n^2)$. The time complexity of Initial population generation algorithm and the fitness function calculation are $O(n_0 n) < O(Tn_0 n^2)$. So the time complexity of genetic algorithm is $O(Tn_0 n^2)$.

The time complexity of the inner loop in Hopfield network is $O(T_1 n^2)$. While T_1 is the number of inner loop iterations. T is the number of outer iterations. So the time complexity of Hopfield network is $O(TT_1 n^2)$.

The time complexity of inner loop in ant colony algorithm is $O(mn^2)$, while m is the number of ants, T is the number of outer iterations. So the time complexity of ant colony algorithm is $O(Tmn^2)$.

Genetic algorithms, Hopfield networks and ant colony algorithm's time complexity are $O(Tn_0 n^2)$, $O(TT_1 n^2)$ and $O(Tmn^2)$, obviously, $O(Tn_0 n^2) > O(TT_1 n^2) > O(Tmn^2)$ .That is, the time complexity of the genetic algorithm the highest, the lowest time complexity is ant colony algorithm.

3.2. Comparison of Space complexity

The distance matrixes of three algorithms must be saved and will takes n2 of the space. Genetic algorithm need to save the population, so it needs $n_0 n$ of the space, under normal circumstances $n_0 > n$, so genetic algorithm space complexity is $O(n_0 n)$. Hopfield network needs to save v,u . v,u are small as n^2 .so the space complexity of Hopfield network is $O(n^2)$. Pheromone matrix and the parameter β of ant colony algorithm are only need n^2 of the space, so the space complexity of ant colony algorithm is $O(n^2)$.

So the time complexity of genetic algorithm, Hopfield networks and ant colony algorithm are $O(n_0 n)$, $O(n^2)$ and $O(n^2)$,and $O(n_0 n) > O(n^2) = O(n^2)$ That is, the space complexity of the genetic algorithm the largest, Hopfield networks and ant colony algorithm have the same space complexity.

3.3. *Comparison results*

3.3.1. *Comparative the merits of the results*

This paper uses TSPLB eil51 to compare the results of the three algorithms. Genetic algorithm uses the following parameters: initial population size n0= 5000, crossover probability p_c = 0.3, mutation probability p_m = 0.01, the maximum number of iterations T = 2000. Value of global optimal is 1007.9. The optimal path length of genetic algorithm's generations is shown in Fig 1

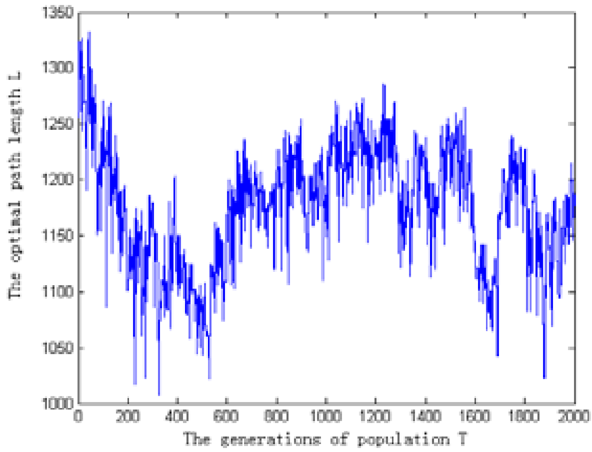

Fig .1 the optimal path length of genetic algorithm's generations

Hopfield network using the parameters as follows:. $A = 500, B = 500, D = 100, \lambda = 0.02, T_1 = 200, T = 100$, and the best path length is 1392.3. The optimal path length of Hopfield network is shown in Figure 2.

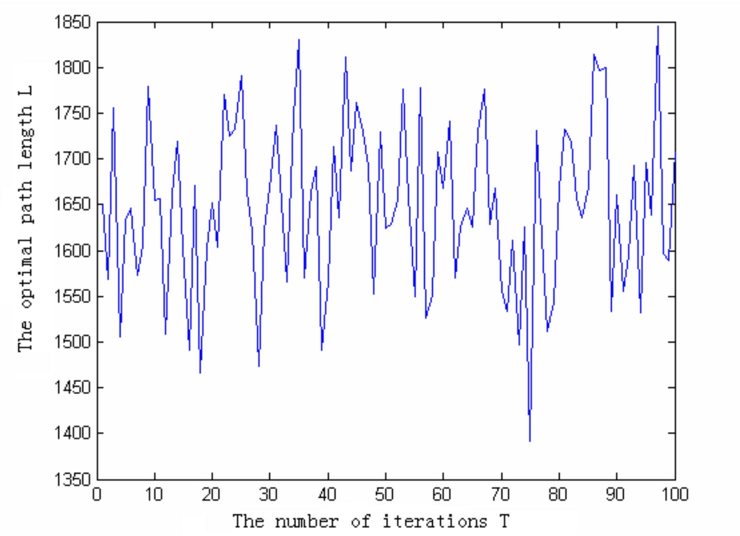

Fig.2 The optimal path length of Hopfield network

The parameters of ant colony algorithm for TSP as follows: $m = 10, T = 500, \alpha = 1, \beta = 5, \rho = 0.1, Q = 600$, $\tau = \frac{1}{D}, \eta = \frac{1}{D}$ and the best path length is 480.13.

The optimal path length of ant colony algorithm is shown in Figure 3

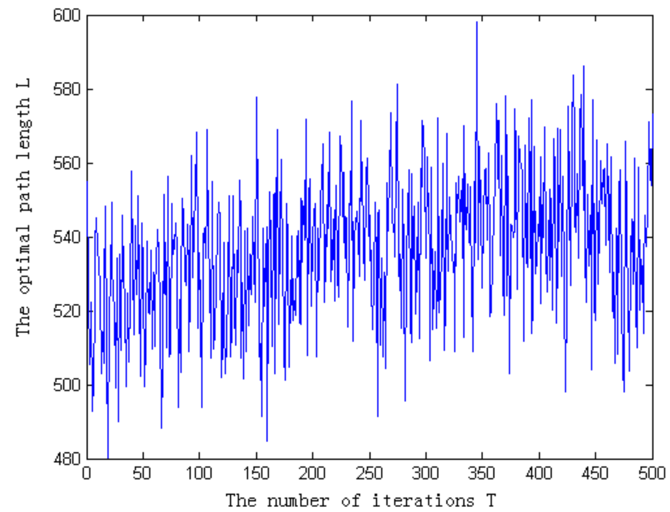

Fig 3 the optimal path length of ant colony algorithm

The optimal solution of eil51 published on TSPLB[7] is 426. The optimal results of genetic algorithms, Hopfield networks and ant colony algorithm were 1007.9, 1392.3 and 480.13.The result of ant colony algorithm is most approach the optimal solution. The results of genetic algorithm and Hopfield network are quite different from the value of the optimal result.

3.3.2. Contrast the partial and overall of the results

Genetic algorithm can ensure the population's evolution by selecting the operator, crossover operator and mutation operator, while selection operator to ensure the best individual parent to survive in a larger probability, crossover probability to use a certain desired cross-breeding to produces a more excellent offspring, mutation operator to a certain mutation probability to the progeny gene mutation expect better offspring.Because of crossover and mutation operators, genetic algorithm can escape local optimum and make it possible to find the global optimum. The result of Hopfield network and ant colony algorithms is good or bad depends on parameter, the value of parameter directly affect the convergence of the algorithm and results of the pros and cons. And these two algorithms are easy to fall into local optimum.

3.4. Comparison the implementation of algorithm

In this paper, we use the code lines of algorithm to reflect the amount of algorithm implementation. The amount of code lines of three algorithms implemented in MATLAB are 186, 93 and 96. From the lines of codes, the most difficult is genetic algorithm.

4. Comprehensive Evaluations

Based on the analysis in Section 3, three algorithms are compared in time complexity, space complexity, the advantages and disadvantages of the calculation results, and difficulty level of realization, etc. We use the application of paired comparison matrix to make comprehensive evaluation, and then give the value of comprehensive evaluation.

The value of a_{ij} in paired comparison matrix from reference [8].

The time complexity, space complexity, the advantages and disadvantages of the calculation results and difficulty level of realization of three algorithms can use comparison to obtain comparison matrix A as follows.

$$\begin{pmatrix} 1 & 5 & \frac{1}{3} & \frac{1}{3} & 7 \\ \frac{1}{5} & 1 & \frac{1}{7} & \frac{1}{7} & 3 \\ 3 & 7 & 1 & 3 & 9 \\ 3 & 7 & \frac{1}{3} & 1 & 7 \\ \frac{1}{7} & \frac{1}{3} & \frac{1}{9} & \frac{1}{7} & 1 \end{pmatrix}$$

The maximum eigenvalue of the matrix is $\lambda_{\max}$ =5.3573. CI is extent of inconsistency of a pair-wise comparison matrix A (n> 1), the formula is:

$$CI = \frac{\lambda_{\max}(A) - n}{n-1} \tag{5}$$

Then we can know CI=0.0893，RI=1.12

$$CR = \frac{CI}{RI} = \frac{0.0893}{1.12} = 0.0798 < 0.1 \quad (6)$$

This shows that A is not a consistent matrix, but A is according with consistency, and the inconsistency of A is acceptable. The maximum eigenvector after normalized of matrix A is $U_A = (0.1679, 0.0534, 0.4619, 0.2861, 0.0307)^Z$ .From this value, we can know when evaluate an algorithm is good or bad. First, we should find the calculation results of different algorithm. Second, we calculate local partial and overall partial of calculation results. Third, we find it`s time complexity and space complexity. Last, we ensure the difficulty level of realization this algorithm. For find comprehensive evaluation on these three algorithms, we analyzer the results base on section 3 can construct the comparison matrix $B_{1\text{-}5}$ from time complexity, space complexity, the advantages and disadvantages of the calculation results, and difficulty level of realization. The value of comparison matrix $B_{1\text{-}5}$as follow:

$$B_1 = \begin{pmatrix} 1 & \frac{1}{3} & \frac{1}{5} \\ 3 & 1 & \frac{1}{3} \\ 5 & 3 & 1 \end{pmatrix}, B_2 = \begin{pmatrix} 1 & \frac{1}{5} & \frac{1}{5} \\ 5 & 1 & 1 \\ 5 & 1 & 1 \end{pmatrix}, B_3 = \begin{pmatrix} 1 & 3 & \frac{1}{5} \\ \frac{1}{3} & 1 & \frac{1}{7} \\ 5 & 7 & 1 \end{pmatrix}, B_4 = \begin{pmatrix} 1 & 3 & 3 \\ \frac{1}{3} & 1 & 1 \\ \frac{1}{3} & 1 & 1 \end{pmatrix}, B_5 = \begin{pmatrix} 1 & \frac{1}{3} & \frac{1}{3} \\ 3 & 1 & 1 \\ 3 & 1 & 1 \end{pmatrix}$$

After test, matrix $B_{1\text{-}5}$ all is right. The weight vector of $B_{1\text{-}5}$ are:

$$U_1 = (0.1047 \quad 0.2583 \quad 0.6370)^Z$$

$$U_2 = (0.0909 \quad 0.4545 \quad 0.4545)^Z$$

$$U_3 = (0.1884 \quad 0.0810 \quad 0.7306)^Z$$

$$U_4 = (0.6000 \quad 0.2000 \quad 0.2000)^Z$$

$$U_5 = (0.1429 \quad 0.4286 \quad 0.4286)^Z$$

They can be considered the point at the time complexity, space complexity, the advantages and disadvantages of the calculation results, and difficulty level of realization of each algorithm. Finally, calculate the total score of each algorithm.

$$P(GA) = 0.17 \times 0.10 + 0.05 \times 0.09 + 0.46 \times 0.19 + 0.29 \times 0.60 + 0.03 \times 0.14 = 0.29$$
$$P(HNN) = 0.17 \times 0.29 + 0.05 \times 0.45 + 0.46 \times 0.08 + 0.29 \times 0.20 + 0.03 \times 0.42 = 0.18$$
$$P(ANN) = 0.17 \times 0.63 + 0.05 \times 0.45 + 0.46 \times 0.73 + 0.29 \times 0.20 + 0.03 \times 0.42 = 0.53$$

Genetic algorithms, Hopfield networks and ant colony algorithm consolidated total scores were 0.29, 0.18, 0.53, that the best algorithm is ant colony algorithm, second is genetic algorithm, third is Hopfield network.

5. Conclusion

This paper presents three intelligent algorithms, namely, basic genetic algorithm, Hopfield neural network and basic ant colony algorithm, solving of the TSP problem in engineering. Then these different kinds of algorithms are compared in time complexity, space complexity, the advantages and disadvantages of the calculation results, and difficulty level of realization, etc. We use the engineering application of paired comparison matrix to make comprehensive evaluation, and then give the value of comprehensive evaluation. This method of comprehensive evaluation is relatively simple and useful.

References

[1]Zhang Jingqang, Cao Yunfu. (2004), A Genetic Algorithm Based on Common Path for TSP [J], Journal of Computer Engineering and Applications, 40[20]:58~61.

[2]Chen Jing, Yang Xiaofan, Zeng Zhi. (2006), A Genetic Algorithm Base on Gene Bank and Multiple-Searching Method for TSP [J],Computer Science，33(8):195~201

[3]Zhao Jianming, Yao Nianmin. (2009), Study on the Efficiency of solving TSP Based on Hopfield Network [j], Science Technology and Engineering ,9(2):437－439

[4]Jiang Guojun. (2001), A modified algorithm of Hopfield network to solve TSP, [j]. Journal o f Zhejiang University (Science Edition), 28(2):160~163

[5]Li Yong, Duan Zhengcheng.(2003), A New Ant System for TSPs [J], Journal of Computer Engineering and Applications, 39(17):103~106

[6]Wang Zhengying, Shi Bingxin.(2001) , Solving QoS Multicast Routing Problem Based on Heuristic Genetic Algorithm [J], Chinese Journal of computer, 24(1):55~61

[7]http://elib.zib.de/pub/Packages/mp-testdata/tsp/tsplib/tsplib.html

[8]Gan Yingai etc,2005, Tsinghua University Press ,Operations Research.