ISSN : 0974 - 7435 Volume 8 Issue 9

BioTechnology
An Indian Journal

FULL PAPER

BTAIJ, 8(9), 2013 [1234-1239]

The performance research on solving TSP by four typical AI algorithms

Yue Wu*, Zheng-qiang Jiang
School of Logistics, Beijing Wuzi University, Beijing 101149, (CHINA)
E-mail: wuyuenet@wo.com.cn; injt19891011@126.com

ABSTRACT

Travelling Salesmen Problem (VRP) has an important theoretical value and practical significance in mathematical and logistics field. It's a typical NP-Hard problem, and artificial intelligent (AI) Algorithm has been already proven to be a very effect way in solving this problem. This paper carried out the performance research on solving TSP by four typical AI algorithms after in-depth analyzed the TSP and these four algorithms (genetic algorithm, particle swarm optimization algorithm, simulated annealing algorithm and ant colony algorithm). This paper verified the TSP solving performance by China travelling salesmen problem experiments and MATLAB programming. The results showed that: considering the average iteration time, SA < PSO< ACA <GA; considering the optimal route length, GA<ACA<SA<PSO; and considering the iterative time to obtain optimal route, SA<ACA<PSO<GA.

KEYWORDS

AI Algorithm;
Travelling salesmen problem;
Solving performance.

INTRODUCTION

Travelling Salesmen Problem (VRP) is always considered as one of the most important theoretical and practical issues in mathematical and logistics field. TSP has a long story which was first introduced by Euler on the knight tour problem[1]. In mathematical field, TSP is a typical NP-Hard problem, and the computation scale grows explosively as the number of nodes increase. In logistics field, TSP is one of the most critical procedures in logistics distribution, and it affects the efficiency and effectiveness in logistics operations directly. So it caught the attention of many experts and scholars in mathematical, operation research, computer science and logistics field as soon as it was proposed, and a lot of effect solving methods has been introduced.

There are plenty of ways to solve TSP, and it can be divided into two parts in general: Accurate solution method and heuristic algorithm. But for large scale TSP, efficient accurate solution method hasn't appeared yet, and the only way to solve this problem is to construct a high quality heuristic algorithm to identify the sub-optimal solution. In current academic, there are a lot of heuristic algorithms used in solving TSP, and AI algorithm has been proven to be effective. AI algorithm imitates and explains certain natural phenomena or process by formatting the stochastic search algorithm by building an internal search mechanism, and it performs well in solving combinatorial optimization and large non-deterministic polynomial complex problems[3,4].

THE DEFINITION AND ITS SOLVING METHODS CLASSIFICATION

The definition of TSP

TSP is a typical combination optimal problem, and it belongs to NP-Hard problem, the conception definition of this issue can be described as: Given a set of n nodes and the distance between each pair of nodes, find a path from node 1 to node with minimal distance, these node can be visit and only visit once. The mathematical description of this issue can be described as: Given a city set $C = (c_1, c_2, \dots, c_n)$ and the distance set between cities $d(c_i, c_j) \in R^+$, find an optimization path $[(c]_{\pi_1}, c_{\pi_2}, \dots, c_{\pi_n})$ from with minimal distance $\min \sum_{i=1}^{i=n-1} [d(c_{\pi_i},]c_{\pi_{(i+1)}}) + d(c_{\pi_n}, c_{\pi_1})$, and the set $(\pi_1, \pi_2, \dots, \pi_n)$ transpose from set (1,2,...,n)[5].

The solving methods classification of TSP

The academic research method on TSP optimization carried out from the following 3 aspects: traditional operation algorithms, local heuristic algorithms and the gradually separated AI algorithms[2].

The first research method used in TSP optimization is traditional operation research algorithm. Among them, Enumeration Algorithm, Lagrangian Relaxation Algorithm, Graph Theory, Dynamic Programming Algorithm, Branch and Bound Algorithm and Cutting Plane Algorithm are the representations[6]. The advantage of these algorithms is capable of accurate solving solutions, but as we know about combination optimal problem, with the increasing of problem scale, these algorithms will gradually out of action (combinatoric explosion). So, as we can see, traditional operation algorithms almost can do nothing in solving large scaled TSP.

The common local heuristic algorithms are Two-Stage Optimization Algorithm, Three- Stage Optimization Algorithm, Steepest Descent Algorithm, Newton Algorithm, and Lin Kernighan (LK) Algorithm. These algorithms have a good effect in calculating local optimal solutions of TSP in a short time, but the disadvantage of these algorithms are easy to fall into local optimal solution[2].

With the developing of computer science, AI algorithm has separated from heuristic algorithm gradually and now it forms an independent system of solving method. Till now, there are many matured AI algorithms, such as Ant Colony Algorithm (ACA), Genetic Algorithm (GA), Simulated Annealing (SA), Tabu Search Algorithm (TS) Neural Network Algorithm (NN) and Particle Swarm Optimization Algorithm (PSO). These algorithms imitate and explain certain natural phenomena or process by formatting the stochastic search algorithm by building an internal search mechanism, which provide a new idea in solving large scaled and complicated problems, although these algorithms cannot guarantee to obtain the optimal solutions in finite time, but the sub-solutions are in the acceptable range after sufficient verifications[7].

Now, the most commonly used algorithms in solving TSP are AI algorithm, among these, GA, PSO, SA, and ACA are most commonly used. Therefore, this paper carried out the performance research on solving TSP by these 4 typical AI algorithms, hoping makes some conclusions.

THE 4 TYPICAL AI ALGORITHMS

Genetic algorithm (GA)

GA was first introduced by Professor John H. Holland and PHD in 1975, this algorithm is inspired by Darwin's biological evolutionary theory, it is a search algorithm based on nature selection, survival of the fittest and genetic idea in biological evolution, it belongs to nonnumeric parallel optimization algorithm[7].

Take the example of TSP, $f(x)$ represents the optimal unction $\min \sum_{i=1}^{i=n-1} [d(c_{\pi_i},]c_{\pi_{(i+1)}}) + d(c_{\pi_n}, c_{\pi_1})$, x is a vector-factor, which represents all the possible values of the variable space. In GA, $f(x)$ generates a fitness function $g(x) = F[f(x)]$ at first, and the fitness function must meet the following two conditions: the range of $g(x), V \in R^+$; $g(x)$ must be maximized when obtain the optimal value. After all these settings, the algorithm can be begun to calculate and the specific

FULL PAPER

steps are listed as follows:

1) Determine the basic parameters, which include the population size $POPSIZE$, crossover probability P_c, mutation probability P_m and some other parameters.
2) Initialization: determine or randomly generate the initiate population with $POPSIZE$ size and calculate the fitness value of each individual G_i;
3) Hybrid: randomly select the chromosome from the population, and exchange them according to the crossover probability P_c;
4) Variation: randomly select the chromosome from the population, and exchange them according to the mutation probability P_m;
5) Reproduction: do reproduction operations by calculating each individual's regeneration numbers, based on the value of corresponding values.
6) Judgment: if the algorithm meets the stopping rule, then return the corresponding individual x of highest fitness value; otherwise, re-execute step 3.

Particle swarm optimization algorithm (PSO)

PSO was first proposed by Kennedy and Eberhart in 1995, it is derived from the study on the foraging behavior of the flock, the algorithm is simple and easy to implement, less adjustable parameters, and has been extensively studied and applied[9]. Similar to GA, PSO is also an iteration based optimization tool, the system initialized with a set of random solutions by iterative search. But the difference between these two algorithms is that PSO has no crossover and mutation operation in its searching process. So, here we don't repeat the specific process steps.

Simulated annealing (SA)

SA was first proposed by Metropolis et al. in 1953[10], and later in 1983, Kirkpatrick et al. first applied it in TSP optimization[11]. It is derived from the study on the annealing process in solid substances. This algorithm iterates from a given solution and generates another solution randomly, and the accept rule allow the object function deterioration in a limited range. The process of this algorithm equals to a "generate new - Judgment - to accept or discard" process. To solving TSP, the specific steps can be described as follows:

1) Initialization: Determine the initiate and end "temperature" T and T_0, annealing speed α and the initiate route $C = C_0$;
2) Inner loop 1: initiate a new rout C_1 within C_0 scope, calculate the corresponding function value of C_0 and, if the corresponding function value of C_1 is smaller, $C = C_1$ or if $\exp\left(-\frac{E(rt)-E(r)}{t}\right) > random(0,1)$, then $C = C_1$; otherwise, $C = C_0$;
3) Inner loop 2: if satisfied the internal circulation sampling stability criteria, then continue next step; otherwise, re-execute step 2);
4) External loop 1: determine new parameters and if the disturbance is accepted, then $C = C_1$; otherwise $C = C_0$;
5) External loop 2: if the new value is accept, then cool down the temperature and $T = \alpha * T_0$, finish one cycle and start the next cycle; otherwise we don't cool down the temperature and ex-execute step 2);

Ant colony algorithm (ACA)

ACA was first proposed by Italian scholar Dorigo M in 1991, it is an imitation of biological ant foraging behavior; it performs well in solving combinatorial optimization and large non- deterministic polynomial complex problems, such as TSP[12]. In solving TSP, the specific steps of ACA are listed as below:

1) Initialization: determine the ant quantity m, heuristic indices α, visibility factor β, pheromone evaporation rate ρ and the total quantity of pheromone Q in one cycle. We make $t = 0$, the corresponding cycle index $nc = 0$, set the maximize cycle index NC_{MAX}, the initiate quantity of pheromone $\tau_{ij}(0) = Constant$ and the $\Delta\tau_{ij}(0) = 0$;
2) Start to iterate: each ant $k\,(k = 1, 2, \ldots, m)$ selects the next node j by probability $P_{ij}^k(t)$ (see in formula 1) and put node j into current solution set;

$$P_{ij}^k(t)=\begin{cases}\frac{[\tau_{ij}(t)]^\alpha * [\eta_{ij}(t)]^\beta}{\sum_{s\in allowed_k}[\tau_{is}(t)]^\alpha * [\eta_{is}(t)]^\beta} & \text{if } j \in allowed_k \\ 0 & \text{otherwise}\end{cases} \quad (1)$$

3) Calculate the length $L_k(k = 1, 2, \ldots, m)$ of each route searched by the ants, record the best current solutions, and modify the pheromone according to formula 2;

$$\tau_{ij}(t+1) = (1-\rho) * \tau_{ij}(t) + \Delta\tau_{ij}(t, t+1) = (1-\rho) * \tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t, t+1) \quad (2)$$

4) For each route (i, j), make $\Delta\tau_{ij}(0) = 0$ and $nc = nc + 1$;

5) Judgment: if $nc < NC_{MAX}$, then re-execute step 2); otherwise, output the best solutions.

NUMERICAL EXAMPLES

Data selection of the examples and algorithms

The parameters of this numerical example have two parts, including example data and algorithm parameters. To example data, we choose the Chinese Travelling Salesman Problem with 31 cities, to algorithms' parameters, this paper first carried out several experiments to determine the best combination of parameters, which listed as follows:

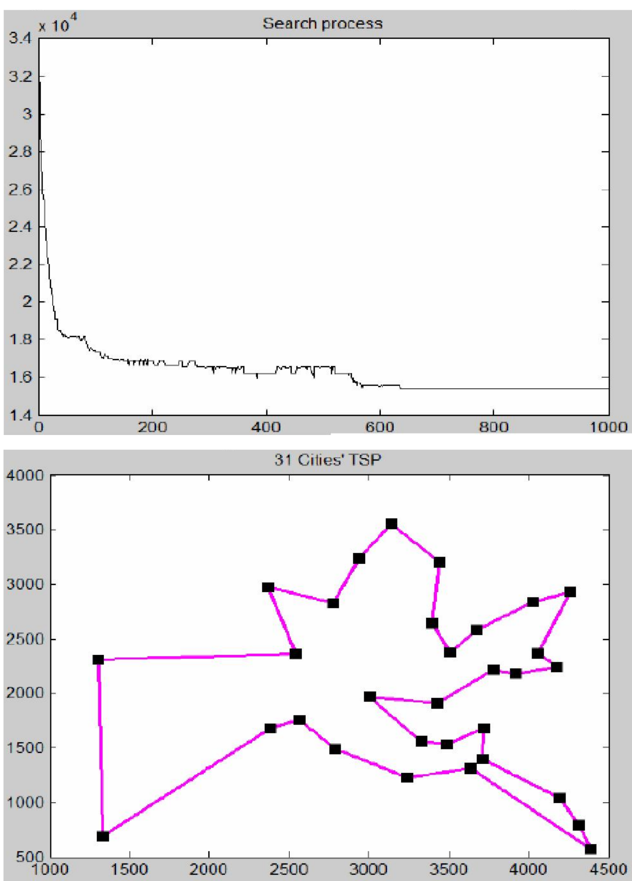

Figure 1 : The solving results diagrams of TSP solved by GA

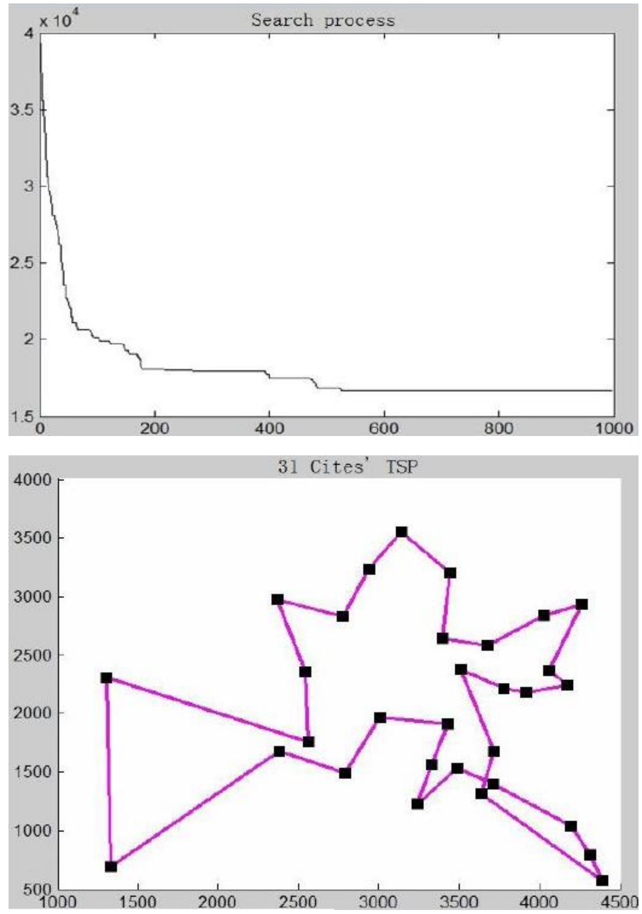

Figure 2 : The solving results diagrams of TSP solved by PSO

To GA: POPSIZE=100, $P_c = 0.85$, $P_m = 0.05$ and $NC_{MAX} = 1000$;

To PSO: SIZE_POP=20, $c_1 = c_2 = 2$, POP_MXA=2, POP_MIN=-2, V_MAX=0.5 and V_MIN=-0.5

To SA: $T_0 = 1000$, $\alpha = 0.9$, $T_{end} = 0.001$ and $L = 200$;

To ACA: $m = 31$, $\alpha = 1$, $\beta = 5$, $\rho = 0.3$, $Q = 100$ and NC_MAX=1000

Example results and analysis

After determine the combination of optimal parameters, this paper carried out 15 times of each algorithm and record the related results, the specific results can be seen as bellows:

(1)To GA

In the 15 times examples, the average time of each experiment is 193.6745 seconds, and the algorithm obtain the optimal solution 15393.4848 kilometers at the 643rd cycle time. So the average iteration time of each cycle is 0.1936745 seconds, and the iteration time need to take to obtain the optimal solutions is 124.5327 seconds. The specific can be seen in figure 1.

(2)To PSO

In the 15 times examples, the average time of each experiment is 76.8732 seconds, and the algorithm obtain the optimal solution 15967.5569 kilometers at the 519rd cycle time. So the average iteration time of each cycle is 0.0768732 seconds, and the iteration time need to take to obtain the optimal solutions is 39.8972 seconds. The specific can be seen in figure 2.

(3)To SA

In the 15 times examples, the average time of each experiment is 15.6141 seconds, and the algorithm obtain the optimal solution 15878.5475 kilometers at the 42rd cycle time. So the average iteration time of each cycle is 0.0156141 seconds, and the iteration time need to take to obtain the optimal solutions is 4.6842 seconds. The specific can be seen in figure 3.

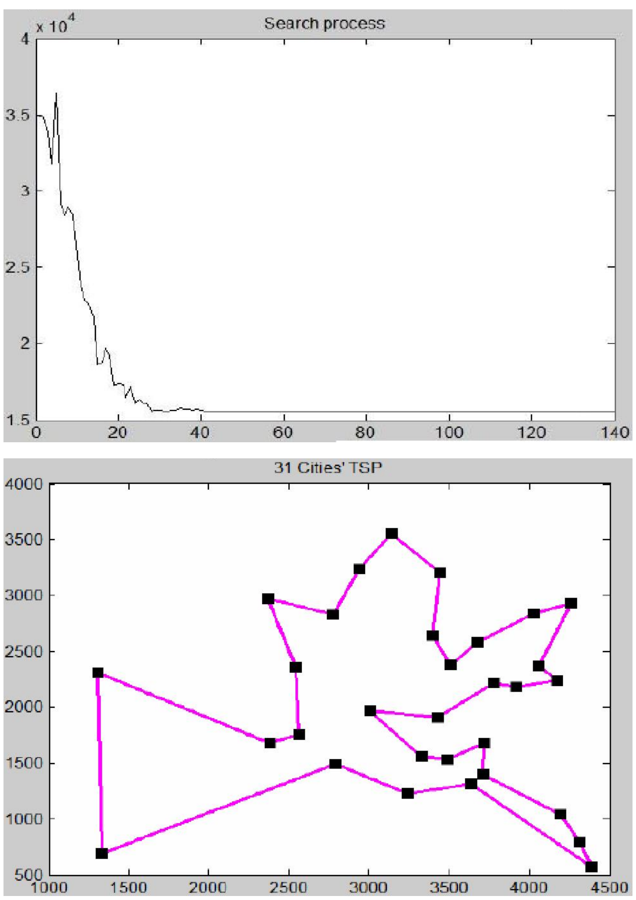

Figure 3 : The solving results diagrams of TSP solved by SA

(4)To ACA

In the 15 times examples, the average time of each experiment is 174.4613 seconds, and the algorithm obtain the optimal solution 15609.4771 kilometers at the 44rd cycle time. So the average iteration time of each cycle is 0.1744613 seconds, and the iteration time need to take to obtain the optimal solutions is 7.6763 seconds. The specific can be seen in figure 4.

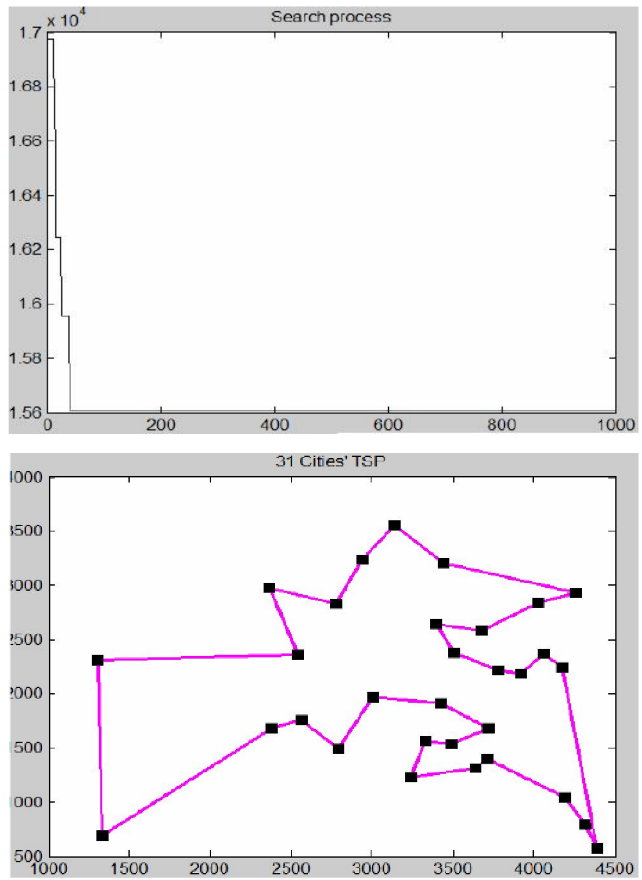

Figure 4 : The solving results diagrams of TSP solved by ACA

TABLE 1 : The performance on solving TSP by these 4 AI algorithm

	Average iteration time Of each cycle (Second)	Optimal route length (KM)	Iteration number when obtain optimal solutions	iteration time when obtain optimal solutions (Second)
GA	0.1936745	15393.4848	643	124.5327
PSO	0.0768732	15967.5569	519	39.8972
SA	0.0156141	15878.5475	42	4.6842
ACA	0.1744613	15609.477	44	7.6763

As we can see from above results, we can integrate the results into TABLE 1 which is listed as below:

As we can see from TABLE 1, considering the average iteration time, SA < PSO< ACA <GA; considering the optimal route length, GA<ACA<SA<PSO; and considering the iterative time to obtain optimal route, SA<ACA<PSO<GA.

CONCLUSIONS

This paper carried out the performance research on solving TSP by four typical AI algorithms after in-depth analyzed the TSP and these four algorithms (genetic algorithm, particle swarm optimization algorithm, simulated annealing algorithm and ant colony algorithm). The paper programs the 4 algorithms and simulates the Chinese Travelling Salesman Problem with 31 cities. The research obtains certain findings, and the results showed that: considering the average iteration time, SA < PSO< ACA <GA; considering the optimal route length, GA<ACA<SA<PSO; and considering the iterative time to obtain optimal route, SA<ACA<PSO<GA. The innovation of this paper is research the performance of these 4 typical AI algorithms and the shortcoming of this paper is with no discussion between the performance and the numerical scale, further research can be extend form this point.

REFERENCE

[1] P.Larranaga, C.M.H.Kuijpers, R.H.Murga et al.; Genetic algorithms for the travelling salesman problem: A review of representations and operators [J]. Artificial Intelligence Review, **13(2)**, 129-170 **(1999)**.

[2] D.L.Applegate, R.E.Bixby, V.Chvatal et al.; The traveling salesman problem: a computational study [M]. Princeton University Press, **(2011)**.

[3] R.E.Korf; Artificial intelligence search algorithms [M]. Chapman & Hall/CRC, **(2010)**.

[4] W.C.Wang, K.W.Chau, C.T.Cheng et al.; A comparison of performance of several artificial intelligence methods for forecasting monthly discharge time series [J]. Journal of hydrology, **374(3)**, 294-306 **(2009)**.

[5] Gao Hai-chang, Feng Bo-qin, Zhu Li; Reviews of the Meta-heuristic algorithms for TSP [J]. Control and Decision, **12(3)**, 241-247 **(2006)**.

[6] Xu Jie-feng; Survey research on Travelling Salesman Problem [J]. Journal of shanghai jiaotong university, **24(4)**, 80-85 **(1990)**.

[7] Shi Z. Advanced artificial intelligence [M]. World Scientific, 2011

[8] L.De Giovanni, F.Pezzella; An improved genetic algorithm for the distributed and flexible job-shop scheduling problem [J]. European journal of operational research, **200(2)**, 395-408 **(2010)**.

[9] R.Eberhart, J.Kennedy; A new optimizer using particle swarm theory [C]//Micro Machine and Human Science, 1995. MHS'95. Proceedings of the Sixth International Symposium on. IEEE, 39-43 **(1995)**.

[10] N.Metropolis, A.W.Rosenbluth, M.N.Rosenbluth et al.; Equation of state calculations by fast computing machines [J]. The journal of chemical physics, **21**, 1087 **(1953)**.

[11] S.Kirkpatrick, M.P.Jr.D.G.Vecchi; Optimization by simmulated annealing[J]. science, **220(4598)**, 671-680 **(1983)**.

[12] A.Colorni, M.Dorigo, V.Maniezzo; Distributed optimization by ant colonies[C]//Proceedings of the first European conference on artificial life., **142**, 134-142 **(1991)**.