Bio-inspired Computing Practice on Traveling Salesman Problem

Wei-jun SU

Department of Mathematics, Gansu Normal University for Nationalities, Hezuo, 747000, China

Keywords: Bio-inspired computing, Traveling salesman problem, Ant colony algorithm, Time complexity, Convergence.

Abstract. Nature always inspires people to find something, people has inspired to proposed many new methods to solution complex optimization problem by various natural phenomenon or process of biosphere, people called the calculation method as Bio-inspired calculation, such as ant colony algorithm, in this paper has analysis the steps of ant colony algorithm and advantages, then given an advise on parameter evaluation, and as a practice example, traveling salesman problem of China 144 city has calculated, show bio-inspired computing calculation is validation and efficient.

Introduction

Review the traveling salesman problem (abbreviated to "TSP") as following: A salesman want to the n cities to sell and faced on how he should choose a route to visit all the cities had the shortest distance traveled? TSP history for a long time, in 1759 Euler had researched a Knight on chessboard who travel around the 64 squares of the chessboard, once and only once visited the 64 squares, and eventual return to the starting point.

TSP has obvious practical significance, in addition, such as transportation, pipelines, route selection, network topology design, postmen deliver letters, and so on, can be abstracted into the TSP to discuss. In 1960, China scholar Guan Mei-gu Professor had give a practice problem: A postman should choose a route can be both traveled the street that he was responsible for the letters to deliver and make the shortest route? Later in academia the so-called Chinese postman problem (abbreviated to "CPP"). In 1960, the A. H. Land and A. G. Doig[1] using branch-and-bound algorithm to solve a lot of the TSP in dozens of cities, another one later made a number of approximate method to solve TSP in dozens of cities or even hundreds of cities. Although at times finding is only approximate. By the year 2004, someone solves 24 978 city TSP, in 2009 it was solved 85 900 cities TSP[2].

On the appearances, CPP and TSP are two similar questions, as long as the carrier must send each location as a "City" on the line. Generally speaking, about successful postman each time want to 200-300 location, if it comes down to the traveling salesman problem to solve, would be a very large problem. Many practical problems can be attributed to the former and also can be attributed to the latter, but both problems there are essential differences in terms of difficulty.

In 1970, the S. A. Cook[3] proposes theory of NP completeness, in 1972, M. Karp[4] tells us that the TSP is NP complete problem, this kind of problem we haven't found an effective algorithm is polynomial algorithm, but CPP there exists polynomial algorithms. Therefore, if the problem can be attributed to TSP and can be attributed to CPP, we suggested generally attributed to CPP so that can solution easily.

Determine whether a problem is NP-complete problem is the main task of the theory of NP completeness, but we must know when a problem is NP complete, we can no longer search for efficient and accurate algorithm (more people would find it impossible to exist a polynomial algorithm), but to find solutions of NP complete problems efficient algorithms in certain special circumstances, or a valid approximation algorithms. In this way, there are two main algorithms for solving TSP, one is guaranteed to get complete algorithm of the optimal solution, typical examples of such enumeration method and branch-and-bound method, cutting plane method, but this algorithm become invalid when the large-scale problems because of combination explode, people had to turn to another kind of approximate algorithms or heuristic algorithm for solving TSP. Bio-inspired
computing algorithms such as Ant Colony Algorithm, Genetic Algorithms, Tabu Search, Hopfield Neural Network and Particle Swarm Optimization Algorithm and immune algorithm can be birthed.

Heuristic algorithm is a kind of special algorithm, its main field is some combination type problem, took an important role in operations research for long period, it usually was understanding for a iterative method, but it does not convergence to optimal solutions of the problem in theory. Because of a quite part combination optimization problem currently is not exists convergence effective algorithm, such that people can accept of a results algorithm that feasible effective and near to the true result, in this case, heuristic algorithm is only method to choose.

**Ant Colony Algorithm**

Ant colony algorithm (abbreviated to "ACA") has inspired by the biological behavior of foraging ants. Biologists have observed that when ants find food, always finding a shortest path from the nest to the food source. After further study found that when searching for food sources, the ants were originally in the vicinity of the nest random exploration, once you have found a food source on the way back to releasing hormone called pheromone. Release pheromones are correlated with food quality information. Many ants in a number of paths in search of food soil left his pheromones. An ant in search of food sources when it encounters a pheromone which will select high pheromone concentration with relatively large probability path so that they form a positive feedback mechanism. Because the ants in a shorter path take off shorter time, so pheromone accumulate faster and less volatile. There remains relatively high pheromone concentration, concentration of pheromone on the path while others were fading as time passes. Lasting a period of time, and eventually the whole ant colony will be able to find a shortest path between the nest and food sources. In 1989 Goss, etc. conduct experiments on the ants of this phenomenon was known as the "2 bridge" experiment.

Experimental results demonstrate that the Ant's ability to find optimal paths. After a period of time, all the ants chose the shortest route. Based on the "2 bridge" experiment, Italy's scholar M. Dorigo, etc. who conducted experiments of a real ant colony in search of food, summed up the ant search mechanisms to collaborate with each other, and in 1991, the famous ant colony optimization algorithm (abbreviated to "ACO").

Due to this strong similarities between the ants has finding the optimal path and the classic solution of TSP, so when M. Dorigo had proposed the ant algorithm and then apply it on the TSP. Artificial ant colony system for the main properties are:

1. the ant colonies are always looking for the lowest cost feasible solutions;
2. each ant has a memory for storing information on its current path, which can be used to construct a feasible solution and evaluation of the quality of solutions, backtracking the path;
3. the current state of the ants can move to any point in the feasible neighborhood;
4. each ant can be given an initial state, but one or more termination conditions;
5. the ant moves from the initial state to a feasible neighborhood, structural solution in a recursive manner, when there is at least one ant satisfy at least one terminated condition, end the construction process;
6. the ant moves to the next neighborhood vertex according to a probabilistic decision rules;
7. when the ant moves to the neighborhood then pheromone trail is updated, this process known as "online pheromone updating in a single step";
8. once to construct a solution, ants along the path to traceback, updating its pheromone trail, this process known as "delayed pheromone update online".

**Basic Steps of ACO**

For example ant colony algorithm for TSP below rationale.

Let m be the number of ants, n represents the scale of the TSP, the number of cities, C be the set of cities. $\tau_0(t)$ are $(i, j)$-section the amount of pheromone on it. $Tabu_k (k = 1, 2, \ldots, m)$ is used to record the
city collection of the ant \( k \) currently had passed through, called tabu table. Then on \( t \) time the probability of the ant \( k \) will moving from \( i \)-city to \( j \)-city is shown as follow.

\[
p_{ij}^k(t) = \begin{cases} 
\frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{j \in \text{allowed}_k} [\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}, & j \in \text{allowed}_k \\
0, & \text{otherwise} 
\end{cases}
\]  
(1)

Among them, \( \text{allowed}_k = C \setminus \text{tabu}_k \) said the ant \( k \) at the next step allows to select a city. \( \alpha \) be the pheromones factor, the importance of the pheromones, its values reflect the importance of pheromone when ants select. \( \eta_{ij} = 1/d_{ij} \), the visibility of the \((i, j)\)-sections, \( d_{ij} (i, j = 1, 2, \ldots, n) \) is the distance between city \( i \) and \( j \). Heuristic factor \( \beta \) for the expectation, expressed the importance of visibility, its values reflect the importance of prior information when ants marching. After a period of time \( T \) when all the ants after a loop completed, pheromone on each \((i, j)\)-sections maybe increased or evaporated, so pheromone needs to be adjusted. Set pheromone evaporation coefficient \( \rho \in (0, 1] \), after each loop pheromone on \((i, j)\)-sections shown as formula:

\[
\tau_{ij}(t + T) = (1 - \rho)\tau_{ij}(t) + \Delta \tau_{ij}(t) 
\]  
(2)

In the formula, \( \Delta \tau_{ij}(t) = \Delta \tau_{ij}^k(t) \), \( \Delta \tau_{ij}^k(t) \) represents ant \( k \) leave the pheromone amount on \((i, j)\)-section in this cycle, \( \Delta \tau_{ij}(t) \) represents the increments of pheromone on \((i, j)\)-section in this cycle.

Ant colony algorithm has following basic steps:

1) The initialization phase. Let current loop \( N_c = 0 \) and time \( t = 0 \); let initializing pheromone on each path \( \tau_{ij}(0) \) be a constant, pheromone increments \( \Delta \tau_{ij}(t) = 0 \); Given the maximum number of loop \( N_{c_{\text{max}}} \), let \( n \) ant on \( n \) vertices(cities). Initialize the \( d_{ij}, \alpha, \beta \) and \( \rho \) parameters.

2) The loop stage. If \( N_c < N_{c_{\text{max}}} \), the loop to continue or end the loop.

3) Each ant \( k \) according to formula (1), move to the next vertex \( j, j \in \text{allowed}_k \)

4) Update the tabu table. Adding the city which each ant pass through in the previous step into the ant's tabu table.

5) According to formula (2) update the amount of pheromone on each path.

6) \( N_c = N_c + 1 \), go back to step (2).

7) The end of the loop, the algorithm terminates and outputs the shortest path.

**Convergence of ACO and Evaluation Parameter**

In the world, the first scholar who research ant colony algorithm convergence problem was Gutjahr, the time complexity of the algorithm is \( O(N_c \cdot N^2 \cdot m) \), if you select \( m = n \), then the time complexity is \( O(N_c \cdot N^3) \). According to algorithm theory, this complexity of the ant colony algorithm is acceptable in computing time.

Currently, through constantly research, people has lots of understand the closely relation that the convergence to the solutions of ant colony algorithm and the problem's structure, and the result of problem usually not global optimal solutions. Second, basic ant colony algorithm has strongly global search capacity, but local search capacity weaker, thus we often needs embedded some specifically or auxiliary skills. As many people think: evolution algorithm just like a repairer, it has only select to adjustment from material which can get, that is, A product of evolution are the result of local optimization in stages, we also found difficulty how to solve global optimization problem from the simple imitation of process, the key to success is through synergies, ensure that the probability of accessibility of any point in state space.

In general, a algorithm in order to have the feature of global optimization, just to satisfy two conditions: (1) has the ability to achieve local optimization (2) from a local optimal state down a better local optimal state transition capability. And ant colony optimization algorithm with the two conditions so that people can look forward to the globally optimal path.
The effect of ant colony algorithm on TSP would be restrained by the effects of parameters, such as $\alpha$, $\beta$, $\rho$, $Q$, etc. Parameter $\rho$ reflected the permanence of pheromone which ant left in its passed track; if the value is too small it means pheromone disappeared too fast, so formed optimal route difficultly, but the value is too big that it has easily fall into local optimal points, therefore, for avoid this two aspects of defects, its suitable value of $\rho$ usually take in about 0.7. The value of $Q$ would take a less effects on algorithm, so we evaluate $Q$ the appropriate value such that it doesn't makes calculating overflow.

When we put the algorithm into practice, the numbers of ant evaluate $n$, and

$$0 < \alpha < 5, 0 < \beta < 5, 0.1 < \rho < 0.99, 1 < Q < 10000.$$  

It is easy to know, the time complexity of this algorithm is $O(NcN^3)$, $Nc$ for the numbers of iteration operation.

**An Example from Chinese 144 Cities TSP**

Since the ant colony algorithm arisen, solving the most problem is a lot of TSP in the famous libraries TSPLIB. Here we solve Chinese 144 cities TSP by the ant colony algorithm, the data from [5], show bio-inspired computing calculation is validation and efficient. Taking $\alpha = 1$, $\beta = 5$ and $\rho = 0.75$, $Q = 1$, numbers of iteration is 10, obtain the shortest route, its length is 3 1245, compare with the known best solution 30351, more than 2.9%, the solution can be acceptable, the shortest path as following:


![TSP tour of 144 cities in the China. The given tour is of length 31245.](figure1)

**Acknowledgement**

This work was supported by President Scientific Research Fund(GSNUXM16-08) and Double Backbone Project Construction plan(GSNU-SHGG-1809) of Gansu Normal University for Nationalities.

**References**


