A hybrid ant colony optimization algorithm based on MapReduce

Ming Cai[1], Yongan Zuo[2,a]

[1]School of Internet of Things Engineering, Jiangnan University,Wuxi 214122,China

[2]School of Internet of Things Engineering,Jiangnan University,Wuxi 214122,China

[a]zuo18262272645@126.com

Keywords: MapReduce, ant colony optimization algorithm, Hadoop, Traveling salesman problem, hybrid ACO

Abstract. The traditional ant colony optimization (ACO) had defects in long searching time and convergence in local optima solution. To solve the problem, a hybrid ant colony optimization algorithm (HACO) was proposed in this paper. Two modes in HACO are defined that are DefaultModel and EliteModel. In order to search the optimum result, the two modes will be automatically switched in this algorithm. It also use the parallel calculation model of MapReduce to loop iteration part of ACO and deploy it on the Hadoop cloud computing platform. Finally, simulation results validate the proposed approach on the traveling salesman problem.

1. Introduction

Vehicle Routing Problem (VRP) mainly through the complex traffic environment and in the many travel programs to find the best travel program, such as minimum with route length or travel time. Face with the current situation of domestic traffic, in order to find a reasonable driving route, is undoubtedly need processing for a huge data transportation. Currently, Developed by the Apache Software Foundation and the open source Hadoop distributed system architecture is very powerful platform for dealing with big date, its main core is a distributed file system (HDFS) to achieve big data storage and computing a MapReduce model[1,8] for parallel computing of massive data.

In recent years, artificial intelligence algorithm has also been rapid development, intelligent algorithms include ant colony algorithm (ACO)[2,3], genetic algorithms, simulated annealing algorithm,etc. Socha and Dorigo proposed one of the now most popular ACO algorithms for continuous domains, called ACOR[4]. It uses a solution archive as a form of pheromone model for the derivation of a probability distribution over the search space. Leguizamón and Coello proposed an extension of ACOR , called DACOR[5], that had the goal of better maintaining diversity during the search.

2. Traditional Ant colony algorithm

A substance called pheromone is released by ants while they have adopted on the route in searching of food. Pheromones will be released by ants in varying degrees according to the length of the route. The study found that the shorter of the route the more pheromones may be released. Other ants will choose a route with most pheromone by the highest probability while also releasing pheromones. However, the pheromone will be volatilize slowly on other routes over time. A positive feedback mechanism is formed to ensure that the route found by ant colony is the most outstanding. Then the ant colony algorithm is firstly proposed (ant colony algorithm, ACO) by Dorigo et in 1991 while is inspired by a foraging behavior of ant colony. It is a probability model algorithm in order to achieve the optimal route found in the figures.

The most important point for the ant colony algorithm is that several parallel route are established at the same time. And in each iteration, the previous information is used to mutually exchange information. In each iteration of the traditional ant colony algorithm, an solution of problem is build step by step by each ant. In this paper to describe the algorithm model for clarity, the classic TSP problem will be used. Set $C = \{c\_1, c\_2, ..., c\_n\}$ represents n cities,

$k(k=1,2,...,m)$ represents the m ants. The m ants are randomly placed in these n cities, and the distance between any city i to j is represented by $d(i,j)$. $\tau_{i,j}$ is the set of pheromone concentration left behind by ants between two cities. Set concentration of pheromone is same between two cities in the initial time. The transition probabilities for ant k moved from city i to j at time t is that:

$$p_{i,j}^{k}(t)=\begin{cases}\dfrac{\tau_{i,j}^{\alpha}(t)\cdot\eta_{i,j}^{\beta}(t)}{\sum_{S\in allowed_k}\tau_{i,j}^{\alpha}(t)\cdot\eta_{i,j}^{\beta}(t)} & ,S\in allowed_k\\ 0 & ,others\end{cases} \tag{1}$$

Wherein, $allowed_k$ represents collection of cities that is the ant k allowed to visit next. α represents the influence level of pheromone when ants choose a route. β represents the influence level of route length when ants choose a route. $\eta_{i,j}(t)=1/d_{i,j}$ is heuristic function which is the expectation when ant k moved from city i to j at time t. So for a ant, the $d_{i,j}$ is shorter the larger $\eta_{i,j}$ will be, and the greater the transition probability $P_{i,j}^{k}$ will be.

In order to avoid the heuristic has no effect because of excess of residual pheromone in the route, so in each iteration after the m ants completion traversal all n cities, next, update the pheromone. Therefore, in the time $t+n$ the pheromone update formula is as follows:

$$\tau_{i,j}(t+n)=(1-\rho)\tau_{i,j}(t)+\rho\cdot\Delta\tau_{i,j}(t) \tag{2}$$

$$\Delta\tau_{i,j}(t)=\sum_{k=1}^{m}\Delta\tau_{i,j}^{k}(t) \tag{3}$$

In the above formula, ρ $(0<\rho<1)$ indicates the degree of volatilization of pheromone. The $1-\rho$ indicates the degree of residual of pheromone. $\Delta\tau_{i,j}(t)$ represents the sum of residual pheromone increment on the route (i,j) in this iteration. Set Initial time is $\Delta\tau_{i,j}(0)=0$. $\Delta\tau_{i,j}^{k}(t)$ indicates a quantity of pheromone residued in the route (i,j) by ant k. Commonly computational model used shown below:

$$\Delta\tau_{i,j}^{k}(t)=\begin{cases}Q/D_k & ,the\ route\ of\ k\ ant\ in\ this\ iteration\\ 0 & ,other\end{cases} \tag{4}$$

Wherein, Q is a constant which influence the speed of convergence to a certain extent. D_K is the length of route by ant k in this iteration.

3. A Hybrid ant colony optimization algorithm based on MapReduce

The traditional ant colony algorithm had defects in long search time and convergence in non-optimal solution. So many researchers put a lot of effort to improve the ant colony algorithm. Based on the combination of DACO and DACOR，This paper proposes a HACO. Different of the ACO is that HACO has two models build route, there are DefaultModel and EliteModel[6]. DefaultModel is deployment an amount of ants in each iteration to construct the route. EliteModel is in each iteration to deploy only one‘elite’ant to construct the route. The ‘‘elite’’ ant select the optimal route R_{best} in the archive R to construct a new solution.

Specifically described as follows:

1): Randomly initialize route and to assess its size of archive is set as k, at the same time selecting the route to build models.

2): If the DefaultModel is selected, N ants will construct probability N new route in each iterations. Set the number of ants is Na. Each ant will create its guided route in the iteration (they seek the route from archive or create a new route). Set a parameter $P_{best}\in[0,1]$ controls probability

of selecting the optimal route from the archive as a guiding route. When probability is $1-P_{best}$, the following two ways is used to get the guiding route. The first one, the guiding route is probabilistically selected from the route archive by their weights. Otherwise, it chosen the i route generated by current ant to be deployed as the guiding route. Once a guiding route generated by each of ants means a new route generated. Na ants will be generated Na routes, and the new routes will be save in the route archive. Then the size of route archive will be $Na+k$. Next, HACO the route archive updates by removing Na worse route. Each newly generated route is compared to the corresponding route to remove the worse one. Finally, a new route archive is generated.

3): For EliteModel, Only a "elite" ant is deployed to find a new route in each iteration. The elite ant selects the optimal route form archive with a probability EP_{best} $\left(0 \le EP_{best} \le 1\right)$. If the newly generated route is better than the optimal route in route archive, it replaces the optimal route.

MapReduce's map function corresponds to the independent solution of each ant which corresponds to a Mapper task. The reduce function is used to acquire a optimum result and update pheromone. Because of the pheromone concentration updating in reducer processing has no effect for next iteration, the result of processing reducer as input of next iteration using the cloud computing concept of channel.

When start the Mapper and Reducer task of MapReduce programming framework, itself will have a certain memory consumption. If each ant as a Mapper task to solving, the solving process is shorter, which consume too much memory. So this paper use the way of hybrid DefaultModel and EliteModel to the operation of MapReduce Mapper.

HACO specific steps are as follows:

1): Initialization phase, MapReduce reads the input files from HDFS between in TSP city node information, and initializes the distance between cities matrix pheromone, α, β, ρ and other parameters.

2): Map phase, Before the beginning of each iteration, the DefaultModel or EliteModel is selected randomly by map function according to proportion setting. The index of ants and optimal solution are the output which is intermediate result expressed as $<key1, value1>$. Where $value1$ is the optimal solution of this map task, and $key1$ is .index of ants. They are stored in a local disk.

3): Reduce phase, the $<key1, value1>$ is a input of reduce function which is the result of Map function. The different inputs put into a collection to obtain the optimal solution. The pheromone updated according to the formula 2. The concentration of pheromone and optimal solution value saved into output file.

The output file acts as a new input file in next iteration. So MapReduce tasks will be linked through the channel technology of cloud calculation.

4. Simulation Analysis and Results

In the experiment, the main HACO algorithm parameters set $\alpha = 1; \beta = 2; Q = 100; \rho = 0.9$; the number of ants is set to 100. Nine nodes are enabled to operate, the number of iterations is $Nc = 100$. Different proportions of DefaultModel and EliteModel is great influence for results which is selected by map function. 10 times experiment was executed with different φ. In this paper, the standard TSP library[7] of test cases ch130 was used as experimental data to verify φ influence of HACO. The result is shown below, the time unit: milliseconds (ms).

Table 1 the influence of result with different φ

The value of φ	time (ms)	optimal solutions
0.1	33073	6543.829
0.2	33105	6527.931
0.3	33142	6502.716
0.4	33243	6483.468
0.5	33487	6251.139
0.6	33561	6046.436
0.7	37491	6032.532
0.8	39329	6023.477
0.9	43295	6017.257

From table 1 can be seen, along with the value of φ from 0.1 to 0.9, the computational complexity is increased by the increasing DefaultModel and the increasing number of ants iterations, and the optimal route will be more accurate, based on the above two sides, the algorithm time had the tendency of increasing and the value of the optimal solution is getting smaller and smaller. In addition, when $\varphi < 0.6$, the difference amongst the optimal solutions are obvious, and the time difference amongst the algorithms is relatively close. Hence, the accuracy of the optimal solution will be sacrificed when the time is chosen less. The phenomenon was presented exactly opposite when $\varphi > 0.6$. The time required by the algorithm increase significantly. But the optimal solution is a tiny difference. If you choose a better accuracy of optimal solution a larger algorithm time will be required as the price. Relatively little time is spent to get a relatively accurate optimal solution when $\varphi = 0.6$. Therefore, this article choose $\varphi = 0.6$.

To validate the algorithm can choose a better route in a relatively less time. In this paper, using standard TSP library of test cases ch130 compare with ACO, ACOR, DACOR.

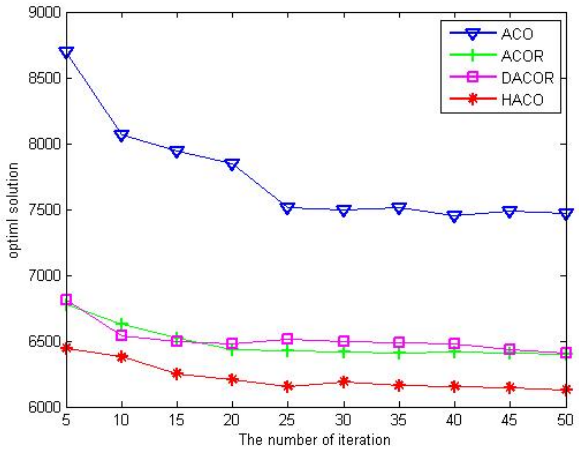

Fig 1. HACO compare with ACO, DACOR, and DACOR in option solution

Red curve represents the optimal solution of HACO n iterations shown in Fig 1. The blue, green, purple curves represent the optimal solution of ACO, DACOR, DACOR respectively. In the figure, we can obviously find that optimal solution obtained by HACO proposed by this paper is better than the other three kinds of algorithms.

5. Summary

This paper use the MapReduce computation model of Hadoop to parallelize the ant colony optimization. The capability of ACO handling big data is enhanced. HACO is proposed which is based on DefaultModel and EliteModel. HACO is significantly improvement the defects in long search time and convergence in non-optimal solution. Throw simulation experiments, HACO can significantly improve performance in handling big data.

References

[1] Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters. Communications of the ACM, Vol 51 (2008)No.1,p.107-113

[2] Dorigo M, Gambardella L M. Ant colony system: a cooperative learning approach to the traveling salesman problem. Evolutionary Computation, IEEE Transactions on, Vol 1(1997) No.1, p.53-66

[3] Dorigo M: Optimization, learning and natural algorithms(Ph. D. Thesis, Politecnico di Milano, Italy 1992)

[4] Socha K, Dorigo M. Ant colony optimization for continuous domains. European journal of operational research, Vol 185(2008)No.3, p.1155-1173

[5] Socha K, Blum C. An ant colony optimization algorithm for continuous optimization: application to feed-forward neural network training. Neural Computing and Applications, Vol 16 (2007)No.3, p.235-247

[6] Liao T, Stützle T, de Oca M A M, et al. A unified ant colony optimization algorithm for continuous optimization. European Journal of Operational Research, Vol 234(2014)No.3, p.597-609

[7] Reinelt G. TSPLIB—A traveling salesman problem library. ORSA journal on computing, Vol 3(1997)No.4, p. 376-384

[8] Zhou Z, Liu G, Su L. A new approach to detect epistasis utilizing parallel implementation of ant colony optimization by MapReduce framework[J]. International Journal of Computer Mathematics, 2015 (ahead-of-print): 1-13.