ISBN 978-981-11-7861-0
Proceedings of 2018 the 8th International Workshop on Computer Science and Engineering (WCSE 2018)
Bangkok, 28-30 June, 2018, pp. 115-120
doi: 10.18178/wcse.2018.06.020

An Improved Ant Colony Algorithm Based on Spark in TSP

Qifan Wu [1+] and Qingsheng Zhu [2]

[1] College of computer science and technology; Chongqing Key Laboratory of Software Theory & Technology; Chongqing University, China

[2] Chongqing Key Laboratory of Software Theory & Technology; Chongqing University, China

Abstract. Traditional ant colony algorithm is easy to fall into local optimum and has slow convergence rate, Moreover, its speed of solving is not satisfactory in above medium scale traveling salesman problem. In order to improve the above problem, a parallel ant colony algorithm based on the coarse granularity of Spark is proposed in this paper. Based on the maximum and minimum ant colony system, this method is implemented in parallel with the Spark framework. At the same time, the following improvements have been made: By using the local search technology with three acceleration functions, the ability of the algorithm to jump out of the local optimal solution is greatly enhanced. Meanwhile, the termination condition of the traditional ant colony algorithm is modified, and the information entropy is used as the criterion of the algorithm convergence judgment, reducing the number of iterations and saving time. The experimental results show that the new parallel ant colony algorithm has a greater improvement than the traditional ant colony algorithm in the solving precision and solving speed.

Keywords: ant colony algorithm, local search, accelerate, Spark, etc.

1. Introduction

Meta heuristic algorithm is an algorithm based on intuition or experience. When it solves problems, it emphasizes satisfaction rather than optimality, so it often obtains satisfactory solution, rather than pursue optimal solution. Typical meta heuristic algorithms include Genetic algorithm [1], particle swarm optimization [2] and ant colony algorithm [3-5].

The ant colony algorithm was first proposed by Italy scholar Marco Dorigo and others in 1991. To find the shortest path between the nest and the food source by simulating ants, with pheromone as a medium to reach the purpose of optimization, so this algorithm is called the bionics algorithm. Ant colony algorithm has the characteristics of positive feedback, distributed and easy to combine with other algorithms [6]. It has a unique advantage in solving the shortest path problem [7-8].

With the rapid development of computer technology and the promotion of the social demand, the problem of "combination explosion" is particularly prominent in today's big data age. The increase of the amount of data explosion leads to this kind of problem. Even using the meta heuristic algorithm, it cannot be efficiently optimized on the ordinary computer. However, most metaheuristic algorithms come from simulating the behaviour of population cooperation and evolution in the nature, which makes the algorithm based on population have natural internal parallelism between individuals. Cloud computing is a new parallel processing technology that has excellent performance in large data computing and network storage [9]. Therefore, the combination of meta heuristic algorithm and cloud computing technology can form a feasible and reliable solution to solve the above problems effectively.

According to the characteristics and shortcomings of the existing algorithm, after considering the search ability and convergence speed of ant colony algorithm, an ant colony optimization algorithm based on cloud

+ Corresponding author. Tel.: + 8613618269651;
E-mail address: qifanwu@live.com .

computing framework Spark is proposed: A local search technique with accelerated function is used for the search path, which greatly improves the possibility of finding the better path and shortens the computing time; The information entropy is used as the criterion to judge the algorithm convergence, and the modification end condition reduces the number of iterations; Finally, the ants are encapsulated in the elastic distributed data set. According to the above improvement, designed the conversion operation of the data set, the ant colony optimization algorithm is parallelized.

2. Improved Ant Colony Algorithm

Based on the maximum and minimum ant colony algorithm, we introduce local search and three speed acceleration, and modify the ending condition of the algorithm. We use information entropy as the criterion of convergence, and reduce unnecessary iterations.

2.1. Adding Local Search

When using the ant colony algorithm to solve the TSP problem, we often find that when finding a better solution, increasing the number of iterations cannot improve the quality of the solution, and the algorithm falls into the local optimum condition. So, we introduce local search strategy. Through the exchange of the edge, the quality of the solution is improved, and the ability of the algorithm to jump out of the local optimal is greatly enhanced. Common local search strategies include k-opt, greedy algorithm and genetic algorithm.

K-opt converts one path into another path based on an exchange heuristic. Given a feasible path, as long as the operation can reduce the length of the current path, the algorithm will operate repeatedly within a given set until any operation cannot generate updates, which results in a local optimal path. Taking 2-opt as an example, the basic principle of the 2-opt method is to replace (i, i+1), (j, j+1) with (i, j), (i+1, j+1), After this transformation, the path of (i+1,… j) in the line is processed in reverse, when the length of the path decreases after the exchange, that is, when the following conditions are met:

$$C_{i,j} + C_{i+1,j} < C_{i,j+1} + C_{j,j+1} \tag{1}$$

Accept the path of exchange, before and after the path changes, as shown in the Figure 1.

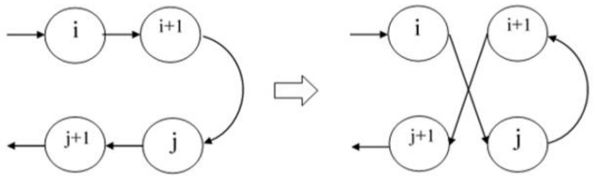

Fig. 1: Switched path.

2.2. Using Three Acceleration Functions

Although the local search improves the quality of the solution and reduces the number of iterations, it needs to search for new paths from the field, and the time cost of the algorithm has increased significantly. So when the size of the city is large, this is obviously an unacceptable price. Using the following three acceleration strategies, the time complexity of the algorithm can be reduced by one level.

1) Each ant in the calculation of how to choose the next city, only calculate the distance of several city nearest the city.

2) When searching neighbourhood, there is no need to try whether a city can connect with all other cities to exchange better results. It only needs to try to exchange with the nearest cities.

3) If we check that the city can't exchange better results, we will set "no longer check" sign. When the unknown city trying to exchange, we will ignore this city.

2.3. The Early End of the Algorithm

In the basic ant colony algorithm, the algorithm termination condition is the maximum iteration number of NMAX, while the value of NMAX is difficult to determine. Because the stopping criterion is not perfect, document [10] is shown that the convergence rate of ant colony algorithm is equivalent to entropy convergence. That is to say, the optimal iteration number of solution convergence is very close to the optimal iteration number of algorithm entropy convergence. Therefore, the information entropy is used as a basis for the convergence judgment of the ant colony algorithm, and it can be used to replace the maximum number of iterations in the algorithm.

In the T iteration of the ant colony algorithm (t=1, 2, …), When the $P_i^{(t)}(i = 1,2,\cdots,m)$,which corresponding to the path length of each ant is generated, the information entropy of the t iteration is as follows:

$$S(t) = -\sum_{i=1}^{m} P_i^{(t)} ln\, P_i^{(t)} \tag{2}$$

The information entropy criterion is:

$$|S(t) - S(t+1)| \times S(t)^{-1} < \varepsilon \tag{3}$$

The ε is the stop threshold of the algorithm, and the range of the value is (0, 0.01).

3. Parallel Ant Colony Algorithm Based on Spark

3.1. Apache Spark Technology

Spark is a big data parallel computing framework based on memory. Because it is based on memory computing, it improves the real-time performance of data processing in big data environment, and ensures high fault tolerance and high scalability. It allows users to deploy Spark on a lot of cheap hardware to form clusters.

3.2. Ant Colony Algorithm Design Based on Spark

In the second part, we improve the design and implementation of ant colony algorithm, and combine Spark parallel computing framework to achieve a parallel algorithm based on improved ant colony algorithm. The design idea of the algorithm is as follows.

1) The algorithm uses a sharing mechanism based on Spark. After the end of each iteration, the updated pheromone matrix is broadcast variable and passed to every machine in the cluster for the next iteration ant colony traversing the city node. In addition, the urban distance matrix also uses broadcast mechanism to transfer it to every node in the cluster, so that ant colony can construct feasible solutions in parallel with all nodes in the cluster.

2) The algorithm encapsulates the ant colony in the RDD set of parallel computing. The Spark program divides the RDD set into several partitions based on the number of cluster nodes, and the RDD in each partition represents one or more ants. Using the functions provided by Spark to design a series of RDD conversion and action operations. Finally, the whole process of constructing a feasible solution by the ant colony represented by the RDD data set is realized.

3.3. Algorithm Flowchart

The specific flow chart of the improved ant colony algorithm based on Spark is shown in the diagram.

In the figure, X represents the number of partitioned Partition, $A_n t_1$, $A_n t_2$, $\cdots$, $A_n t_m$ says m different ants, $P_1, P_2, \cdots, P_m$represents the optimization path for the M ants.

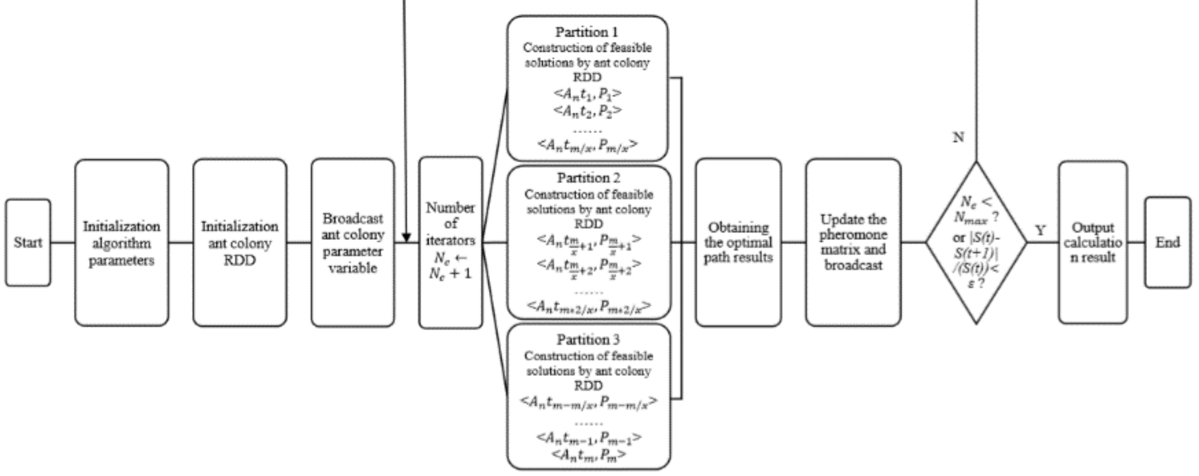
Start
Initialization algorithm parameters
Initialization ant colony RDD
Broadcast ant colony parameter variable
Number of iterators $N_c \leftarrow N_c + 1$
Partition 1 Construction of feasible solutions by ant colony RDD
$<A_n t_1, P_1>$
$<A_n t_2, P_2>$
……
$<A_n t_{m/x}, P_{m/x}>$
Partition 2 Construction of feasible solutions by ant colony RDD
$<A_n t_{\frac{m}{x}+1}, P_{\frac{m}{x}+1}>$
$<A_n t_{\frac{m}{x}+2}, P_{\frac{m}{x}+2}>$
……
$<A_n t_{m*2/x}, P_{m*2/x}>$
Partition 3 Construction of feasible solutions by ant colony RDD
$<A_n t_{m-m/x}, P_{m-m/x}>$
……
$<A_n t_{m-1}, P_{m-1}>$
$<A_n t_m, P_m>$
Obtaining the optimal path results
Update the pheromone matrix and broadcast
$N_c < N_{max}$? or $|S(t)-S(t+1)|/(S(t))< \varepsilon$?
N
Y
Output calculation result
End

Fig. 2: Flowchart.

4. The Experiment and the Result Analysis

The test case is from the international TSP case library TSPLIB, and ch150, kroA200, lin318 and d493 four sets of data are selected to carry out the simulation experiment. The distance between cities is kept decimal, and the best solution, average solution and run time of 10 experiments are recorded. The operating parameters, the number of ants m=50, α =1 β =5, ρ =0.02. The attribute information of the node in the experimental environment is detailed in Table 1.

Table 1: Node information table

Project	Value
Number of nodes	6
Spark version	1.6.3
Spark mode	Stand Alone
JDK version	1.7.0
Node CPU	Intel Xeon CPU 2.80 GHz*16
Node memory	2GB

First test the smaller size of the city group, use ch150 to carry out the experiment, the data obtained as shown in Table 2. The optimal solution can be found by Spark-MMAS and MMAS, and the quality of the Spark-MMAS mean solution is better than that of MMAS, but the average time is almost two times that of the first two algorithms. The reason for this phenomenon is that the Spark computing framework is small in scale, the resource preparation required by cluster initialization, and the communication cost between clusters account for a large part, so it is not as fast as the other two algorithms of single version.

Table 2: Comparison results of several algorithms in the ch150

Algorithm	Known optimal solution	Optimum solution	Mean solution	Mean time (s)
AS	6528	7012	7249	1481
MMAS	6528	6528	6641	1462
Spark-MMAS	6528	6528	6579	3012

The size of the city is increased, and the kroA200 and the lin318 are selected for the experiment. The data obtained are shown in Table 3 and Table 4. The average solution of Spark-MMAS is better than the first two algorithms. Spark-MMAS spends roughly the same time on the kroA200 problem as the first two algorithms. In the lin318 problem, the time is less than half of AS and MMAS. The power of parallelization algorithm is beginning to show.

Table 3: Comparison results of several algorithms in the kroA200

Algorithm	Known optimal solution	Optimum solution	Mean solution	Mean time (s)
AS	29368	33504	33895	3190
MMAS	29368	29616	30143	3157
Spark-MMAS	29368	29426	30012	3470

Table 4: Comparison results of several algorithms in the lin318

Algorithm	Known optimal solution	Optimum solution	Mean solution	Mean time (s)
AS	42029	46153	46738	11339
MMAS	42029	43426	43827	11240
Spark-MMAS	42029	42588	42686	5420

When the size of the city continues to increase, the d493 test algorithm is selected and the data obtained are shown in Table 5. Although the Spark-MMAS cannot achieve the optimal solution at this time, the quality of the average solution is obviously superior to that of AS and MMAS. With the expansion of the scale of the problem, the average solution of AS and MMAS has a obvious decline compared with the known optimal solution. In addition, the cost of AS and MMAS is five times that of Spark-MMAS in time.

Table 5: Comparison results of several algorithms in the d493

Algorithm	Known optimal solution	Optimum solution	Mean solution	Mean time (s)
AS	35002	40605	40850	38462
MMAS	35002	37220	37717	38216
Spark-MMAS	35002	36538	36912	7943

Based on the above test data, we can conclude that our algorithm has stronger ability of searching global optimal solution than traditional ant colony algorithm and MMAS algorithm. It improves the quality of algorithm solution and accelerates the speed of algorithm, and shows better performance.

5. Conclusion

On the basis of analysing the principle and realization method of ant colony algorithm, according to the algorithm is easy to fall into local optimum, it is difficult to determine the algorithm repeatedly times and when faced with large scale city the operation becomes slow, an improved ant colony algorithm based on Spark is proposed. The algorithm uses the Spark computing framework. Parallel computing is used to speed up operation. Meanwhile, in search path, local search strategy with three acceleration functions is combined to improve the ability of jumping out of local optimum and also save time. Finally, the convergence of information entropy is used to modify the ending condition of the algorithm, which reduces unnecessary iteration and saves time. The experimental results show that our algorithm can effectively improve the quality of solution and reduce the running time when dealing with large scale cities, which proves the effectiveness and feasibility of the algorithm. In the future study and research, we will further improve the algorithm, expand the application field of the algorithm, and solve more practical problems.

6. Acknowledgements

This work is supported by National Nature Science Foundation of China (61272194, 61502060) and Project of Chongqing Education Commission (KJZH17104).

7. References

[1] Goldberg D E. Genetic algorithms in search, optimization, and machine learning. *Upper Saddle River: Addison Wesley Professional.* 1989

[2] Kennedy J and Eberhart R C Particle swarm optimization. *Proc. of the First IEEE International Conference on Neural Networks.* 1995, pp1942-1948.

[3] Dorigo M and Gambardella L M. Ant colonies for the travelling salesman problem *J. Biosystems.* 1997, 43(2) pp73-81.

[4] Dorigo M and Gambardella L M. Ant colony system: a cooperative learning approach to the traveling salesman problem *J. IEEE Transactions on Evolutionary Computation.* 1997, 1(1) pp53-66.

[5] Dorigo M and Maniezzo V. Ant system: Optimization by a colony cooperating Agents *J. Man and Cybernetics part B, IEEE Transactions on Systems.* 1996, 26(1) pp29-41.

[6] WU Hua-feng and CHEN Xin-qiang. On the Ant Colony Algorithm based on Natural Selection Strategy to Solve TSP Problem. J. Journal of Communi-cations. 2013, 34(04) pp165-170.

[7] LIN Dong-mei, WANG Dong and LI Ya. A Hybrid Algorithm for Solving Multi-traveling Salesman Problem Biodegradation. *J. Journal of Computer Applications.* 2011, 28(08) pp2876-2879.

[8] SUN Kai and WU Hong-xing. Ant Colony and Particle Swarm Optimization Algorithm for Solving TSP Problem. *J. Computer Engineering and Applications.* 2013, 48(34) pp60-63.

[9] Zhang J, Gu Z and Zheng C.Survey of research progress on cloud computing *J. Application Research of Computers.* 2010, 27(2) pp 429-433.

[10] Wang CB. *Entropy convergence analysis and application of ant colony algorithm* (China: Sichuan Normal University). 2009