Computers and Electrical Engineering 000 (2016) 1–10

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Cooperative ant colony-genetic algorithm based on spark☆

Dong Gaifang, Fu Xueliang*, Li Honghui, Xie Pengfei

Inner Mongolia Agricultural University, Hohhot 010018, China

ARTICLE INFO

Article history:
Received 14 July 2016
Revised 28 September 2016
Accepted 28 September 2016
Available online xxx

Keywords:
Spark
Mapreduce
Ant colony optimization
Genetic algorithm
TSP

ABSTRACT

By taking full advantages of both the map and reduce function for the MapReduce parallel framework and the memory computation for the Spark platform, this paper designs and implements the algorithms for solving the traveling salesman problem based on ant colony algorithm on MapReduce framework and Spark platform. Next, adds the nearest neighbor selection strategy for choosing next city for the Spark platform ant colony algorithm, and combines it with genetic algorithm by using the optimal individual between ant colony algorithm and genetic algorithm, in order to update each other's best individual at the end of each iteration. Experimental results show that with the increase of ant colony size, compared to the stand-alone ant colony algorithm, MapReduce ant colony algorithm reflects the superiority of parallel computation; compared to the MapReduce ant colony algorithm, Spark platform ant colony algorithm reflects the superiority of memory computing. Cooperated with genetic algorithm, the solution has been improved significantly in its precision.

© 2016 Elsevier Ltd. All rights reserved.

Combinatorial optimization problem [1] is an important branch of operations research, which involves sorting, filtering and other issues. The main objective is to find the optimal solution from the feasible solution set. Although the definition of combinatorial optimization problem is very simple, solving the optimal solution is very difficult. Due to that the solving process needs huge storage space and very long running time, so traditional computer is unlikely to be achieved, that is, the so-called "combinatorial explosion" phenomenon. Therefore, for the combinatorial optimization problem, how to use heuristic algorithm to solve the problem becomes the focus of attention. Heuristic algorithm is summarized according to the nature of some excellent population collaborative behavior. The outstanding representatives of this kind of algorithm are ant colony optimization [2–4], genetic algorithm (GA) [5] and particle group optimization algorithm (PSO) [6] etc.

Ant colony algorithm is a kind of bionics algorithm which is put forward to simulate the natural ants foraging behavior. Now it has been widely used in fault recognition [7], vehicle routing problem [8,9], the system identification [10], data mining [11,12], image processing [13–15], and other fields. The basic principle of ant colony algorithm is that when ants search path, it will secrete a certain amount of pheromones. Ant colony always communicate through the path pheromones in order to achieve the goal of optimal path. After the end of each iteration, the ant colony algorithm will volatilize pheromone on the path to form a positive feedback mechanism and to ensure that the ant colony can find the optimal path results. Since it appeared, there have been a large number of scholars putting forward the improvement strategies. However, in the treatment of medium and large scale TSP, there is still problem of long searching time and easily falling into local optimal solution [16]. Many researchers focus on the improvement of the shortcomings of the ant colony algorithm, such as easy stagnation, slow convergence speed and so on. Gambardella and other scholars proposed the Ant-Q algorithm, which can be

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Guest Editor Dr. Z. Zheng.

* Corresponding author.
E-mail addresses: fxliang@126.com, fuxl@imau.edu.cn (F. Xueliang).

http://dx.doi.org/10.1016/j.compeleceng.2016.09.035
0045-7906/© 2016 Elsevier Ltd. All rights reserved.

a good way to maintain the balance between knowledge discovery and knowledge utilization in the process of constructing the solution [17]. WJ Gutiahr provides a graph search ant system (Ant System Graph-based, GBAS), which has a certain probability of convergence to the optimal solution of the problem [18]. Wu Bin and Shi Zhongzhi proposed a meeting algorithm, which effectively improved the quality of the ant first traversal algorithm. Zhang Xuliang, Zhang Jinbin and Zhuang Changwen introduced the cooperative mechanism in basic ant colony algorithm, and proposed the enhanced ant colony algorithm based on cooperative learning. On the convergence of the ant colony algorithm, Hou Yunhe did a lot of related research and achieved some preliminary research results; Ding Jianli and Sun Xi combined genetic algorithm and ant colony algorithm integration with the theory of Markov stochastic process for the convergence of ant colony algorithm and made a lot of fruitful research.

To get rid of the defect of long searching time, in this paper, we design and implement the basic ant colony algorithm based on MapReduce parallel framework (MRACO). Compared with basic ant colony algorithm in the single node environment, the computational time has been greatly improved. While Spark is a parallel platform which is more suitable for iterative computation in recent two years. Therefore, we design the basic ant colony algorithm based on Spark platform (Spark-ACO). Experimental results show that the Spark-ACO takes less computational time than MRACO. In view of the defects of the ant colony easily falling into the local optimal solution, Spark-ACO is improved: on the one hand, the nearest neighbor strategy is used for ants to traverse the city node; on the other hand, ant algorithm and genetic algorithm are combined to improve the defects of local optimum for ant colony. The experimental results show that the improved hybrid algorithm has a great improvement on the accuracy of the solution.

The second section of the paper describes the implementation of basic ant colony algorithm; the third section discusses the design of ant colony optimization based on MapReduce (MRACO); in the fourth section, ant colony algorithm based on Spark (Spark-ACO) and its improvement have been explained; the fifth section is the experimental results and analysis, and the sixth section is the conclusion and discussion.

1. Ant colony optimization algorithm

The traveling salesman problem (TSP) is described as follows: A traveling businessman will visit n cities. He starts from one city, and visit the cities one by one-each city can only be visited once, at last, he returns to the starting city. The shortest path is required to traverse all the cities.

The mathematical model of ant colony algorithm for solving TSP problem can be described as follows: suppose there are n nodes in different cities, the m ants (k= 1,2,..., m) are put into different cities randomly selected. Each ant does a complete traversal of all the cities, returns to the starting city, and records the total length of the path and path length; During the ant colony searching the optimal path process, each ant has a tabu list $tabu_k(k=1,2,\ldots,m)$ to record the cities which have been passed. Every step of the ants must be based on the state transition probability $P^k{}_{ij}(t)$. $P^k{}_{ij}(t)$ indicates that the probability between the city j and the city i for ant k at t time, as shown in the formula (1):

$$\mathrm{p}_{\mathrm{ij}}^{\mathrm{k}}(t) = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(ij) \times \eta_{ij}^{\beta}}{\sum\limits_{s \in allowed_k} \tau_{ij}^{\alpha}(ij) \times \eta_{ij}^{\beta}} & if\ j \in allowed_k \\ 0 & otherwise \end{cases} \tag{1}$$

$allowed_k$ in the formula (1) is the collection of cities which ant k allows to access at present; $\tau_{ij}(t)$ is the residual pheromone between city i and city j at time t; $\eta_{ij}(t)$ is the degree of expectation city i moves to the city j at time t, the value of $\eta_{ij}(t)=1/d_{ij}$, d_{ij} is path length between city i and j; α represents the information heuristic factor; β represents an expected heuristic factor.

In the end of each iteration, all the paths between the cities will be residual amount of pheromone. If not promptly volatilizing the pheromone, too much pheromone will be left on the path in the next iteration process. So the randomness of the ants selecting the path will be increased. Adjust the pheromone according to the formula (2) and (3), as shown below:

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

$$\tau_{ij}(t) = \sum_{k=1}^{m} \tau_{ij}^{k}(t) \tag{3}$$

ρ is the information pheromone evaporation coefficient in the formula (2) and (3) which range is (0,1), 1 - ρ is the pheromone residual factor; $\tau_{ij}(t)$ is the pheromone between city i and city j at time t; $\Delta\tau_{ij}(t)$ is the pheromone increment between city i and city j from t to $t+n$ time; $\tau^k{}_{ij}(t)$ is the pheromone for ant k generated at t to $t+n$ time between the city i and city j. When solving the TSP problem, it is better to use the Ant-Cycle model to update the pheromone on the path, so the paper uses the Ant-Cycle model, as shown in the formula (4):

$$\tau_{ij}^{k}(t) = \begin{cases} \frac{Q}{L} & <i,j> \in Path_L \\ 0 & otherwise \end{cases} \tag{4}$$

Download English Version:

https://daneshyari.com/en/article/4955183

Download Persian Version:

https://daneshyari.com/article/4955183

Daneshyari.com