Expert Systems with Applications 38 (2011) 3873–3883

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Parallelized genetic ant colony systems for solving the traveling salesman problem

Shyi-Ming Chen *, Chih-Yao Chien

Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan, ROC

ARTICLE INFO

Keywords:
Traveling salesman problem
Ant colony systems
Genetic algorithms
Parallelization
Parallelized genetic ant colony systems

ABSTRACT

In this paper, we present a new method, called the parallelized genetic ant colony system (PGACS), for solving the traveling salesman problem. It consists of the genetic algorithm, including the new crossover operations and the hybrid mutation operations, and the ant colony systems with communication strategies. We also make an experiment with three classical data sets got from the TSP library to test the performance of the proposed method. The experiment results show that the performance of the proposed method is better than Chu et al.'s method (2004).

1. Introduction

The traveling salesman problem (TSP) (Lawler, Lenstra, & Shmoys, 1985) is a NP-complete problem. It tries to find a complete route with a minimal cost. Because the traveling salesman problem is a good background for testing the performance of optimization methods, some methods (Adachi & Yoshida, 1995; Baraglia, Hidalgo, & Perego, 2001; Chien & Chen, 2009; Ellabib, Calamai, & Basir, 2007; Fiechter, 1994; Freisleben & Merz, 1996; Hopfield & Tank, 1985; Kirkpatrick, Gelatt, & Vecchi, 1983; Liu & Zeng, 2009; Martin & Otto, 1996; Naimi & Taherinejad, 2009; Saadatmand-Tarzjan, Khademi, Akbarzadeh-T, & Moghaddan, 2007; Xie & Liu, 2009; Yi, Bi, Yang, & Tang, 2008) have been developed for solving the traveling salesman problem. Kirkpatrick et al. (1983) presented a simulated annealing method to solve the traveling salesman problem by simulating the annealing action in metallurgy. Hopfield and Tank (1985) presented a method based on neural networks to make routing decisions for solving the traveling salesman problem by mimicking the properties of biological neurons. Fiechter (1994) presented a tabu searching method with a parallelized mechanism to solve the traveling salesman problem. Adachi and Yoshida (1995) presented the parallelized techniques and the idea of protected chromosomes to reduce the searching space to speed up the processing time of genetic algorithm (GA) for solving the traveling salesman problem. Freisleben and Merz (1996) presented a hybrid method combining the GA and local search heuristics to find a near-optimum solution for solving the traveling salesman problem. Martin and Otto (1996) presented a method combining the simulated annealing (SA) with the local heuristics to solve the traveling salesman problem. Baraglia et al. (2001) presented a hybrid heuristic genetic algorithm for solving the traveling salesman problem. Ellabib et al. (2007) presented a method with exchange strategies for multiple ant colony systems to solve the traveling salesman problem. Saadatmand-Tarzjan et al. (2007) presented a novel constructive-optimizer neural network for solving the traveling salesman problem. Yi et al. (2008) presented a fast elastic net method for solving the traveling salesman problem. Naimi and Taherinejad (2009) presented an ant colony algorithm using a local updating process to solve the traveling salesman problem. Chien and Chen (2009) presented a method based on parallelized genetic ant colony systems to solve the traveling salesman problem. Liu and Zeng (2009) presented a method using genetic algorithms with reinforcement learning to solve the traveling salesman problem. Xie and Liu (2009) presented a method using multiagent optimization systems for solving the traveling salesman problem.

Swarm intelligence is an important research topic for solving optimization problems. It is inspired by the social behaviors of real insects or animals. Ant colony optimization (ACO) is the most popular one in this research topic. The original algorithm of ACO is known as the ant system (Dorigo, 1992) which was proposed by Dorigo to solve the traveling salesman problem. Since then, some algorithms based on the ACO are presented, such as the ant colony system (ACS) (Dorigo & Gambardella, 1997a, 1997b), the MMAS (Stützle & Hoos, 1996, 2000), the rank-based AS (Bullnheimer, Hartl, & Strauss, 1997) and KCC-Ants (Naimi & Taherinejad, 2009). These algorithms are all based on the idea of updating the pheromone information to search the shortest route. ACO algorithms are also used to solve other combinational optimization problems, such as the quadratic assignment (Stützle & Hoos, 2000) and the job scheduling (Merkle, Middendorf, & Schmeck, 2002; Merkle & Middendorf, 2003).

Genetic algorithms (GA) are important techniques in the research area of evolutionary computing. Because GA can search

* Corresponding author. Tel.: +886 2 27376417; fax: +886 2 27301081.
E-mail address: smchen@mail.ntust.edu.tw (S.-M. Chen).

0957-4174/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2010.09.048

the domain space globally, it has been widely used to solve large-scale combinational optimization problems. As long as the fitness function is determined, we can obtain a good result by means of GA. Adachi and Yoshida (1995) presented protected chromosomes techniques to implement GA to solve the traveling salesman problem.

In this paper, we propose a new method, called the parallelized genetic ant colony system (PGACS) to solve the traveling salesman problem. First, we use the ant colony system to obtain the initial solutions. Then, we use the initial solutions as the initial population of GA to obtain better solutions. If GA searches a better solution, we use the global update of ACS to feedback the learning experience to ACS. After a certain number of cycles, we use the communication strategies (Chu, Roddick, & Pan, 2004) to exchange the learning experience between groups to speed up the convergence process. The performance of the proposed method is better than Chu et al.'s method (2004).

The rest of this paper is organized as follows. In Section 2, we briefly review the concepts of ant system (AS) and the ant colony optimization (ACO). In Section 3, we briefly review the concepts of genetic algorithms. In Section 4, we present a new method to solve the traveling salesman problem by parallelized genetic ant colony systems. In Section 5, we show the experiment results of the proposed method based on three classical TSP data sets. The conclusions are discussed in Section 6.

2. Ant colony optimization

In this section, we briefly review basic concepts of ant systems (AS) and the ant colony optimization (ACO). The ant system was proposed by Dorigo et al. (Colorni, Dorigo, & Maniezzo, 1991; Dorigo, 1992; Dorigo, Maniezzo, & Colorni, 1992, 1996). It is a cooperative population-based searching algorithm and is inspired by the foraging behavior of real ants. When a real ant searches the food from its nest, it will form a route based on the quantities of the pheromone on each edge in which it passed through. Each ant will deposit the pheromone on each traveled edges while searching, such that the pheromone level of those visited edges will be increased and the pheromone level on unvisited edges will be decayed. The other ants will search their own routes according to their experiences based on the pheromone levels in the edges that they choose to travel. The higher the pheromone level on an edge, the more the attractive to an ant. The more the ants passed the edge, the higher the pheromone level on this edge. The procedure of this searching behavior can be applied to solve the travel salesman problem. Assume that there are n cities and m ants and assume that the initial pheromone level on each edge is set to a very small non-zero positive constant τ_0. In each cycle, each ant starts at a randomly selected city and visits the rest of each city once and only once according to the transition rule based on the pheromone level on each edge. Therefore, the learning procedure is to update the level of pheromone. The ideas of ant systems to solve the traveling salesman problem are reviewed as follows (Dorigo and Gambardella, 1997; Dorigo et al., 1996):

(1) Transition rule: From city r, the next city s in the route is selected by ant k among the unvisited cities memorized in J_r^k according to the following rule:

$$s = \arg\max_{u \in J_r^k} \left[\tau(r,u) \cdot \eta(r,u)^{\beta}\right], \quad \text{if } q \leqslant q_0 \quad \text{(Exploitation)}$$

or select the next city s with the transition probability $P_k(r,s)$,

$$P_k(r,s) = \begin{cases} \frac{\tau(r,s)\cdot\eta(r,s)^{\beta}}{\sum_{u \in J_r^k} \tau(r,u)\cdot\eta(r,u)^{\beta}}, & \text{if } s \in J_r^k, \\ 0, & \text{otherwise}, \end{cases} \quad \text{if } q > q_0 \quad \text{(Bias exploitation)}$$

where J_r^k is the set of cities that remain to be traveled by the kth ant, $\tau(r,u)$ is the pheromone level between city r and city u, $\eta(r,u)$ is the inverse of the Euclidian distance from city r to city u, and the parameter β is the relative importance of the pheromone level versus the Euclidian distance.

(2) Pheromone update rule:

$$\tau(r,u) \leftarrow (1-\rho)\cdot\tau(r,s) + \sum_{k=1}^{m} \Delta\tau_k(r,s),$$

$$\Delta\tau_k(r,s) = \begin{cases} \frac{1}{L_k}, & \text{if } (r,s) \in \text{ the route performed by the } k\text{th ant}, \\ 0, & \text{otherwise}. \end{cases}$$

The process of AS for solving the TSP is shown in Fig. 1 (Colorni et al., 1991). An ant system requires a lot of computation time to process the transition probability. In order to improve the searching efficiency, some improved algorithms are proposed, such as the MAX–MIN ant system (MMAS) (Stützle & Hoos, 1996, 2000), the ant colony system (ACS) (Dorigo & Gambardella, 1997a, 1997b; Gambardella & Dorigo, 1996), the rank-based AS (Bullnheimer et al., 1997) and KCC-Ants (Naimi & Taherinejad, 2009).

ACS (Dorigo & Gambardella, 1997a, 1997b; Gambardella & Dorigo, 1996) is based on AS, but it updates the pheromone by two operations, i.e., the local update and the global update. The local update operation is performed after the ants moved to the next city, where its main idea is to diversify the searching space performed by the subsequence ants during the cycle. The global update is performed only for the shortest route after all ants complete their routes, where its main idea is that only the best ant could contribute to the colony. The update rules of ACS are shown as follows (Dorigo and Gambardella, 1997):

(1) Local pheromone update:

$$\tau(r,s) \leftarrow (1-\rho)\cdot\tau(r,s) + \rho\cdot\tau_0,$$

where ρ is the pheromone evaporation parameter and $0 < \rho < 1$.

Procedure Ant System Algorithm for Solving the TSP:

Step 1: Set parameters and initialize the pheromone on each edge.

Step 2: For each ant, choose the next city with the probability $P_k(r, s)$ given by the transition rule. If each ant travels all cities, go to **Step 3**. Otherwise, go to **Step 2**.

Step 3: Update the pheromone on each edge.

Step 4: If the termination condition met, then print the best route. Otherwise, go to **Step 2**.

Fig. 1. The process of AS for solving the TSP (Colorni et al., 1991).