Information Sciences 180 (2010) 3182–3191

Contents lists available at ScienceDirect

Information Sciences

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

A swarm intelligence approach to the quadratic minimum spanning tree problem

Shyam Sundar, Alok Singh *

Department of Computer and Information Sciences, University of Hyderabad, Hyderabad 500 046, India

ARTICLE INFO

Article history:
Received 2 October 2009
Received in revised form 9 March 2010
Accepted 1 May 2010

Keywords:
Artificial bee colony algorithm
Constrained optimization
Heuristic
Quadratic minimum spanning tree problem
Swarm intelligence

ABSTRACT

The quadratic minimum spanning tree problem (Q-MST) is an extension of the minimum spanning tree problem (MST). In Q-MST, in addition to edge costs, costs are also associated with ordered pairs of distinct edges and one has to find a spanning tree that minimizes the sumtotal of the costs of individual edges present in the spanning tree and the costs of the ordered pairs containing only edges present in the spanning tree. Though MST can be solved in polynomial time, Q-MST is $\mathcal{NP}$-Hard. In this paper we present an artificial bee colony (ABC) algorithm to solve Q-MST. The ABC algorithm is a new swarm intelligence approach inspired by intelligent foraging behavior of honey bees. Computational results show the effectiveness of our approach.

1. Introduction

The quadratic minimum spanning tree problem (Q-MST) is an extension of the well known minimum spanning tree problem (MST) in graphs. In Q-MST, costs are associated not only with edges of the graph but also with ordered pairs of distinct edges and the objective is to find a spanning tree that minimizes the sumtotal of the costs associated with individual edges and the costs resulting from ordered pairs consisting of those edges present in the spanning tree. Formally, let $G = (V, E)$ be a connected undirected graph, where V denotes the set of nodes and E denotes the set of edges. Given a non-negative cost function $w : E \rightarrow \mathbb{R}^+$ associated with edges of G and a non-negative cost function $c : (E \times E - \{(e, e),\ \forall e \in E\}) \rightarrow \mathbb{R}^+$ associated with ordered pairs of distinct edges, the Q-MST problem seeks a spanning tree $T \subseteq E$ that minimizes

$$\sum_{e_1 \in T} \sum_{\substack{e_2 \in T \\ e_2 \neq e_1}} c(e_1, e_2) + \sum_{e \in T} w(e)$$

Q-MST has several practical applications. It occurs when transferring oil from one pipe to another in a situation where the cost depends on the type of interface between two pipes. This quadratic cost structure also arises in the connection of overground and underground cables or in a transportation or road network with turn penalties. The presence of quadratic costs in all these cases leads to the minimum spanning tree problem with quadratic cost instead of the usual linear cost [16].

Q-MST problem was introduced and proved $\mathcal{NP}$-Hard by Asad and Xu [1,15]. Due to the $\mathcal{NP}$-Hard nature of Q-MST, any exact method is not practical even for moderately large instances and one has to look for heuristics.

Assad and Xu [1,15] proposed a branch-and-bound based exact method and two heuristic for the Q-MST and applied them on graph instances with number of nodes between 6 and 15. Even on these small instances, the solutions obtained

* Corresponding author. Tel.: +91 4023134011.
E-mail addresses: mc08pc17@uohyd.ernet.in (S. Sundar), alokcs@uohyd.ernet.in (A. Singh).

0020-0255/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2010.05.001

by the two heuristics were far from optimal. Zhou and Gen [16] proposed a genetic algorithm based on Prüfer encoding [11] for the Q-MST problem and observed that this genetic algorithm is superior to two heuristic algorithms of [1,15] on random instances with number of nodes between 6 and 50. Soak et al. [14] proposed edge-window-decoder encoding and showed that a genetic algorithm based on this encoding performs much better than genetic algorithms based on other encodings on Q-MST and other problems. All algorithms were tested by Soak et al. on Euclidean instances with number of nodes between 50 and 100.

In this paper, we have proposed an artificial bee colony algorithm (ABC) to solve Q-MST. The ABC algorithm is a new swarm intelligence technique based on the intelligent foraging behavior of honey bees. This technique was proposed by Karaboga [2]. We have compared our ABC approach with the best approaches. Computational results demonstrate the effectiveness of our approach.

The rest of this paper is organized as follows: Section 2 provides a brief introduction to the artificial bee colony algorithm. Section 3 describes our ABC approach for Q-MST. Computational results are reported in Section 4, whereas Section 5 contains some concluding remarks.

2. Artificial bee colony algorithm

The artificial bee colony (ABC) algorithm is a new swarm intelligence technique inspired by intelligent foraging behavior of honey bees. The first framework of ABC algorithm mimicking the foraging behavior of honey bee swarm in finding good solutions to combinatorial optimization problems was presented by Karaboga [2]. Further developments in the ABC Algorithm have been carried out by Karaboga and Basturk [3–7], Singh [13] and Pan et al. [9]. On the basis of their foraging behaviors, real bees are classified into three groups – employed, scouts and onlookers. All bees which are presently exploiting a food source are termed "employed". The employed bees bring loads of nectar from the food source to the hive and may share the information about their food sources with onlooker bees. "Onlookers" are those bees that wait in the hive for employed bees to share information about their food sources. Those bees, which are presently searching for new food sources in the neighborhood of the hive, are termed "scouts". Employed bees share information about their food sources by dancing in a common area in the hive called dance area. The nature and the duration of a dance depends on the nectar content of the food source currently being tapped by the dancing bee. Onlooker bees observe numerous dances before selecting a food source. The onlookers have a tendency to select a food source according to a probability proportional to the nectar content of that food source. Therefore, good food sources attract more bees than the bad ones. Whenever a scout or a onlooker finds a food source it becomes employed. Whenever a food source is exhausted fully, all those employed bees, which are currently exploiting it, abandon it and become scouts or onlookers. Therefore, employed bees and onlookers bees do the job of exploitation, whereas exploration is left to scouts.

Karaboga [2] modeled this intelligent foraging behavior of real honey bee swarm into artificial bee colony (ABC) algorithm to solve real world optimization problems. In ABC algorithm model, Karaboga also classified the colony of artificial bees into same three groups – employed, onlookers and scouts. First half of the colony consists of employed bees, whereas the latter half consists of onlookers. In ABC algorithm, each food source represents a candidate solution to the problem. Each employed bee is associated with a unique food source, and, therefore in ABC algorithm, the number of employed bees is equal to the number of food sources. The nectar amount of a food source is a function of the quality (fitness) of the candidate solution being represented by that food source.

ABC algorithm is an iterative algorithm. It starts by associating each employed bee with a randomly generated food source (solution). Then, during each iteration, each employed bee determines a new neighboring food source of its currently associated food source and computes the nectar amount (fitness) of this new food source. If the nectar amount of this new food source is higher than that of its currently associated food source, then this employed bee moves to this new food source abandoning the old one, otherwise it continues with the old one. When this process is completed by all employed bees, then they start sharing information about their food sources with onlookers. Each onlooker selects a food source probabilistically according to the nectar amount (fitness) of that food source. The probability p_i of selecting a food source i is computed as:

$$p_i = \frac{f_i}{\sum_{j=1}^{k} f_j}$$

where f_i is the fitness of the candidate solution associated with the food source i and k is the total number of food sources. This selection method is known by the name "roulette wheel selection method" in genetic algorithm community. In roulette wheel selection method, candidate solutions are selected from the population randomly, with their probability of selection proportional to their relative fitness in the population. Thus, fitter candidate solutions have a greater chance of survival than the weaker ones. Therefore, rich food source attracts more onlookers than the poor ones. Once all onlookers have selected their food sources in the aforementioned way, each of them determines a new neighboring food source of its selected food source and computes the nectar amount of this new food source. Among all the neighboring food sources determined by onlookers associated with a particular food source i and food source i itself, best food source will be the new location of food source i. If a candidate solution represented by a food source i does not improve for a predetermined number of iterations, then the solution represented by food source i is considered to be explored fully and hence food source i is presumed to be exhausted and its associated employed bee abandons it to become scout. This scout is transformed into an employed bee