Expert Systems with Applications 39 (2012) 5681–5686

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

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

A hybrid algorithm based on tabu search and ant colony optimization for k-minimum spanning tree problems

Hideki Katagiri *, Tomohiro Hayashida, Ichiro Nishizaki, Qingqiang Guo

Faculty of Engineering, Hiroshima University, 1-4-1 Kagamiyama, Higashi-hiroshima, Hiroshima 739-8527, Japan

ARTICLE INFO

Keywords:
k-Minimum spanning tree
Tabu search
Ant colony optimization
Hybrid algorithm
Approximate solution

ABSTRACT

This paper considers an efficient approximate algorithm for solving k-minimum spanning tree problems which is one of the combinatorial optimization in networks. A new hybrid algorithm based on tabu search and ant colony optimization is provided. Results of numerical experiments show that the proposed method updates some of the best known values with very short time and that the proposed method provides a better performance with solution accuracy over existing algorithms.

1. Introduction

A k-minimum spanning tree (k-MST) problem is one of combinatorial optimization problems formulated in networks, and the objective of the problem is to find a subtree with exactly k edges such that the sum of the weights attached to edges is minimal, called k-minimum spanning tree. The k-MST problem is a generalized version of minimum spanning tree (MST) problems; when $k = |V| - 1$ where $|V|$ is a cardinality of vertices in a graph, the k-MST problem corresponds to the MST problem. The wide varieties of decision making problems in the real world can be formulated as k-MST problems, e.g. telecommunications (Garg & Hochbaum, 1997), facility layout (Foulds, Hamacher, & Wilson, 1998), open pit mining (Philpott & Wormald, 1997), oil-field leasing (Hamacher, Jörnsten, & Maffioli, 1991), matrix decomposition (Börndorfer, Ferreira, & Martin, 1997, 1998) and quorum-cast routing (Cheung & Kumar, 1994).

The k-MST problem was firstly introduced by Hamacher et al. (1991). Since the k-MST problem is NP-hard (Fischetti, Hamacher, Jörnsten, & Maffioli, 1994; Marathe, Ravi, Ravi, Rosenkrantz, & Sundaram, 1996), it is difficult to solve large-scale problems within a practically acceptable time. Therefore, it is very important to construct approximate solution methods which quickly find a near optimal solution (Jörnsten & Løkketangen, 1997; Katagiri, Hayashida, Nishizaki, & Ishimatsu, 2010). Blum and Blesa (2005) proposed several approximate solution methods including metaheuristics such as evolutionary computation, ant colony optimization (ACO) and tabu search (TS). They compared their performances through benchmark instances (KCTLIB, 2003) and showed that an ACO approach is the best for relatively small ks, whereas a TS-based approach has an advantage for large ks with respect to solution accuracy. Recently, it is shown that both ACO and TS are promising approaches to solving not only k-MST problems but also other combinatorial optimization problems (Abdallah, Emara, Dorrah, & Bahgat, 2009; Aladag, Hocaoglu, & Basaran, 2009; Li, Zhu, Yan, & Yan, 2010; Öncan, Cordeau, & Laporte, 2008; Tseng, Lin, & Huang, 2008). Moreover, in order to construct more efficient solution method by combining some effective approximate ones, hybrid algorithms (Chen, Pan, & Wu, 2008; Chen & Chien, 2011; Liu & Zeng, 2009; Naimi & Taherinejad, 2009) have been highly attracted attention to many researchers.

In this paper, we propose a new hybrid approximate solution algorithm based on TS and ACO. In order to demonstrate efficiency of the proposed solution method, we compare the performances of the proposed method with those of existing algorithms using the well-known benchmark instances (KCTLIB, 2003) that are easily accessible through the internet. The numerical experimental results show that the proposed method updates some of the best known solutions and values with very short computational times, and that the proposed method provides a better performance with solution accuracy over existing algorithms.

2. Problem formulation and existing solution methods

Given that a graph $G = (V, E)$ where V is a set of vertices and E is a set of edges, a subtree with exactly k edges, called k-subtree T_k, is defined as

$$T_k \in G, \quad k \leqslant |V| - 1.$$

Then a k-MST problem is formulated as

$$\begin{aligned} \text{minimize} \quad & \sum_{e \in E(T_k)} w(e) \\ \text{subject to} \quad & T_k \in \mathcal{T}_k, \end{aligned}$$

where $\mathcal{T}_k$ is the set of all possible k-subtrees T_k in G, $E(T_k)$ denotes the edges of T_k and $w(e)$ is a weight attached to an edge e. The above

* Corresponding author.
E-mail address: katagiri-h@hiroshima-u.ac.jp (H. Katagiri).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.11.103

problem is to seek a k-subtree with the minimum sum of weights. If the problem size is small, the problem can be easily solved by finding an optimal solution after enumerating all possible k-subtrees in a given graph. If the size of problem is not so large, it can be solved by some exact solution algorithm such as a branch and bound method (Cheung & Kumar, 1994) and a branch and cut algorithm (Freitag, 1993). However, it has been shown that the k-MST problem is NP-hard even if the edge weight is in {1,2,3} for all edges, or if a graph is fully connected. The problem is also NP-hard for planar graphs and for points in the plane (Marathe et al., 1996). Therefore, it is important to construct not only exact solution methods but also efficient approximate solution methods.

3. Proposed algorithm

Since TS Glover and Laguna (1997) is an extension of local search, its intensification ability, which means the ability of robustly finding a very good local optimal solution in relatively narrow search space, is very high. However, the diversification capability of TS, which means the ability of exploring a wide solution space and covering the whole region to be searched, is relatively lower. This characteristic of tabu search reflects the fact that the TS-based algorithm by Blum and Blesa is the best for solving the benchmark instances of k-MST problems in the case of large ks.

On the other hand, ACO-based algorithm by Blum and Blesa can seek a very good solution for k-MST problems with small ks, which we think is caused by its high diversification capability. In fact, as will be described later in the experimental results, it is observed that ACO by Blum and Blesa often finds better "best" values than their TS method, and that some of the "best" values are better than even the proposed method. However, the objective function values obtained through ACO may quite-variable due to its lower intensification ability than TS.

Realizing that TS and ACO are considered to be complementary to each other, in this paper, we propose a hybrid algorithm that achieves a balance between the diversification and the intensification by incorporating the ideas of ACO into a TS algorithm.

The outline of the proposed algorithm is as follows:

Step 1 (*Generation of an initial solution*). For a vertex selected at random, the application of Prim method is continued until a k-subtree is constructed. Let the obtained k-subtree be an initial solution and the current solution T_k^{cur}.
Step 2 (*Initialization of parameters*). Initialize the tabu lists and the values of parameters such as tabu tenure tl_{ten} and aspiration criterion levels.
Step 3 (*Tabu search-based local search procedure*). Search the neighborhood based TS, and store a set of local minimum solutions. If the current tabu tenure tl_{ten} is greater than tt_{max}, go to Step 4. Otherwise, return to Step 2.
Step 4 (*Ant colony optimization-based diversification procedure*). Expand the exploration area based on ACO to increase the diversity of the solutions.
Step 5 (*Terminal condition*). If the current computational time is greater than *TimeLimit*, terminate the algorithm. Otherwise, return to Step 2.

Let T_k^{cur}, T_k^{gb} and T_k^{lb} be the current solution, the best found solution and local optimum solution, respectively. Then, we describe the details on the procedures in Steps 3 and 4.

3.1. Tabu search-based local search

In this section, we describe the details on the TS-based local search algorithm performed in Step 3.

For a set $V(T_k)$ of vertices included in k-subtree T_k, we define

$$V_{NH}(T_k) := \{v|\{v, v'\} \in E(G), v \notin V(T_k), v' \in V(T_k)\}.$$

Let T_k^{NH} be a local minimum solution of k-subtree obtained by adding $v_{in} \in V_{NH}(T_k)$ to T_k and deleting $v_{out} \in V(T_k)$. Then, the neighborhood of T_k denoted by $NH(T_k)$ is defined as a whole set of possible T_k^{NH} in G.

In the proposed local search algorithm, the next solution through transition is selected as the k-subtree that has the best objective function value of all solutions $T_k^{NH} \in N(T_k^{cur})$ as follows:

$$T_k^{NH_{best}} := \arg \min_{T_k^{NH} \in NH(T_k^{cur})} \{f(T_k^{NH})\}.$$

It should be stressed here that our neighborhood is different from the Blum–Blesa one. While only the leaf vertices can be selected in the transition of the Blum–Blesa's algorithm, all of vertices adjacent to the current tree can be selected in the proposed algorithm. This extension enables us to strengthen the intensification ability of local search, but the computational time for finding the best neighborhood solution may be longer. One of the promising approaches to decreasing the computational time is to incorporate the solution algorithm for minimum spanning tree (MST) problems, which has an advantage that it is solved in a polynomial time. However, a direct application of the MST algorithms by Prim and Kruskal is not efficient even if it is a polynomial-time algorithm because so large number of applying the MST algorithm is needed. Realizing such difficulty, we employ a more efficient method of obtaining the best neighborhood solution without applying the MST algorithm, which will be described later in the details of the algorithm.

When using a local search algorithm, there is a problem of how to go out of a local optimal solution or how to avoid cycling among a set of some solutions. In order to resolve such a problem, we use two tabu lists *InList* and *OutList*, which keep the induces of removed edges and added edges, respectively. A tabu tenure, denoted by θ, is a period for which it forbids edges in the tabu lists from deleting or adding. In details, at the beginning, we set an initial value of the tabu tenure tl_{ten} to tt_{min} which is the minimum tabu tenure defined as

$$tt_{min} := \min\left\{\left\lfloor\frac{|V|}{20}\right\rfloor, \frac{|V|-k}{4}, \frac{k}{4}\right\}.$$

Let nic_{int} be the period of the best found solution T_k^{gb} not being updated. If $nic_{int} > nic_{max}$, then tabu tenure is updated as $tl_{ten} \leftarrow tl_{ten} + tt_{inc}$, where

$$nic_{max} := \max\{tt_{inc}, 100\}, \quad tt_{inc} := \left\lfloor\frac{tt_{max} - tt_{min}}{10}\right\rfloor + 1.$$

If the current tabu tenure tl_{ten} is greater than tt_{max} defined as

$$tt_{max} := \left\lfloor\frac{|V|}{5}\right\rfloor,$$

the local search algorithm is terminated, and diversification strategy based on ACO is performed.

When checking whether the transition from the current solution to some solution in T_k^{NH} is acceptable, if an edge e in *InList* or *OutList*, which is related to the transition as the added edge or deleted edge, satisfies the condition $\gamma_e > f(T_k^{NH})$, then the transition is permitted. The parameter γ_e called *aspiration criterion level* is given to all of edges and is initially set to

$$\gamma_e = \begin{cases} f(T_k^{cur}), & e \in E(T_k^{cur}), \\ \infty, & e \notin E(T_k^{cur}). \end{cases} \tag{1}$$

In each explored solution T_k, γ_e is updated as $\gamma_e \leftarrow f(T_k)$ for every $e \in E(T_k)$.