Discrete Applied Mathematics 181 (2015) 310–315

Contents lists available at ScienceDirect

Discrete Applied Mathematics

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

Note

Maintaining centdians in a fully dynamic forest with top trees

Hung-Lung Wang *

Institute of Information and Decision Sciences, National Taipei University of Business, Taipei, 100, Taiwan

ARTICLE INFO

Article history:
Received 27 September 2012
Received in revised form 2 September 2014
Accepted 22 September 2014
Available online 7 October 2014

Keywords:
Fully dynamic forest
Centdian
Top tree
Nonlocal search

ABSTRACT

In this paper, we consider the problem of maintaining the centdians in a fully dynamic forest. A forest is said to be fully dynamic if edge insertions, edge deletions, and changes of vertex weights are allowed. Centdian is a specific kind of facility that integrates the notions of center and median by taking a convex combination on the objective functions of both problems. This work extends the results in Alstrup et al. [2] within the same time complexity, i.e., linear time preprocessing and $O(\log n)$ per update, where n is the number of vertices of the components being updated.

1. Introduction

A forest is a graph with each component being a tree. A forest is said to be fully dynamic if edges can be both inserted and deleted and vertex weights can be changed. Under such a dynamic setting, information, like the minimum spanning tree and the diameter of each component, changes with the evolution of the environment. The issue of manipulating dynamic forests has widely been investigated [1,3–5,11,12]. In this paper, we focus on maintaining a specific facility, called *centdian*, in a fully dynamic forest. The notion of centdian is proposed by Halpern [6–8], which is an integration of the notions of *center* and *median*. Both centers and medians are "locations" that minimize some given objective functions, where a location stands for either a vertex or a point on an edge. In this paper, we focus on manipulating locations being vertices. For the center problem, the objective is to find a vertex with minimum *eccentricity*, where the eccentricity of a vertex is the longest distance from any other vertex to it. For the median problem, the objective is to find a vertex that minimizes the sum of distances from all the other vertices to it (sum of distances for abbreviation). Halpern generalized these two notions by taking a convex combination on the two objectives, i.e., finding a vertex which minimizes a convex combination on the eccentricity and the sum of distances.

For the dynamic center and median problems in a forest, the best known results are given by Alstrup et al. [1], which are of linear time preprocessing and $O(\log n)$-time per update, where n is the number of vertices of the components being updated. They developed the technique of *nonlocal search* on *top trees*. Top tree is a data structure designed for manipulating a fully dynamic forest. It extends the idea of *topology tree*, proposed by Frederickson [5], with no degree constraint on the vertices. Based on top trees, nonlocal search is developed for searching objects with nonlocal properties. A property is said to be local if the object satisfies that in all subtrees it is contained in; otherwise, it is nonlocal. For example, "being a diameter of a tree" is local since a path P which is a diameter of a tree is a diameter of each subtree which P is contained in. However, "being a center of a tree" is nonlocal since for a tree center v, one can find a subtree T' containing v, in which v is not a center. An example is given in Fig. 1.

* Tel.: +886 2 2322 6504; fax: +886 2 23226293.
E-mail addresses: hlwang@ntub.edu.tw, dweeblan@gmail.com.

http://dx.doi.org/10.1016/j.dam.2014.09.013

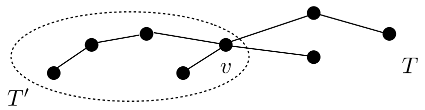

Fig. 1. A tree T with unit edge lengths. Vertex v is a center of T but not a center of T'.

Maintaining dynamic centdians can be reduced to the problem of maintaining dynamic medians, based on an observation given by Halpern [6, Theorem 2]. Halpern proved that a centdian of a tree T is the median of another tree obtained by applying an appropriate transformation on T. Nevertheless, in practice, transformations on the input cause overheads and are usually avoided. In this paper, we show how nonlocal search can be performed on the top tree of the input tree T, without any transformation on T. In the rest, we define the problem formally and summarize some properties of a center, a median, and a centdian in Section 2. A brief review of top trees and nonlocal search is given in Section 3.1, and the design of the function that enables nonlocal search is shown in Section 3.2. Finally, concluding remarks are given in Section 4.

2. Preliminaries

In this section, we define the notation and the problem investigated in this paper. For any graph-theory terms not defined here, please refer to Harary [9]. For any graph G, we use $V(G)$ and $E(G)$ to denote its vertex set and edge set, respectively. An edge of end vertices u and v is denoted by uv. For any subgraph H of G, we use $G - H$ to denote the complement of H with respect to G. For any vertex v, we use $G - v$ to denote the subgraph of G induced by $V(G) \setminus \{v\}$. Similarly, for any edge uv, we use $G - uv$ to denote the spanning subgraph of G with edge set $E(G) \setminus \{uv\}$. The input of the dynamic centdian problem is a forest $F = T_1 \cup T_2 \cup \cdots \cup T_k$, where T_i, $1 \leqslant i \leqslant k$, is a component of F. The vertices $V(F)$ and edges $E(F)$ are weighted by $w\colon V(F) \to \mathbb{R}^+$ and $\ell\colon E(F) \to \mathbb{R}^+$, respectively.

Definition 1 (*Fully Dynamic Forests*)**.** A forest F is said to be fully dynamic if the following three operations are allowed:

- link(x, y): insert edge xy, where $x \in T_i$, $y \in T_j$, and $i \neq j$.
- cut(e): delete edge e from F.
- set-w(x, γ): change the weight of vertex x to γ.

Remark. In [1], an operation called expose is also introduced. expose takes two vertices u and v as the arguments and make them the *boundary vertices* (defined later in Section 3.1). It is used when retrieving information of a vertex. The application of expose does not affect the location of the facilities we investigate in this paper. We defer the detailed usage of expose to [1].

Let $T = (V, E)$ be a component that is in manipulation. For any two vertices x and y, the unique path from x to y is denoted by $P[x, y]$, and the distance from x to y is defined as $d(x, y) = \sum_{e \in P[x,y]} \ell(e)$. In the case where T is rooted, we use T_x to denote the subtree rooted at vertex x and $\hat{T}_x$ be $T - T_x$. For succinctness, the notation of a tree is also used to denote its vertex set. The weight $w(x)$ of the vertex x is abbreviated as w_x, and the sum of weights of the vertices in a subtree T' is denoted by $w(T') = \sum_{x \in T'} w_x$. Three kinds of facilities discussed in this paper are defined as follows.

Definition 2 (*Center, Median, and Centdian*)**.** A center of T is a vertex minimizing f_c, where $f_c(x) = \max_{y \in V} d(x, y)$. A median of T is a vertex minimizing f_m, where $f_m(x) = \sum_{y \in V} w_y d(x, y)$. For a given $\lambda \in [0, 1]$, a centdian is a vertex minimizing f_λ, where $f_\lambda(x) = \lambda f_m(x) + (1 - \lambda) f_c(x)$.

In the following, a center, a median, and a centdian of the given tree T are referred to as v_c, v_m, and v_λ, respectively. Now we are ready to define the *dynamic centdian problem*.

Definition 3 (*The Dynamic Centdian Problem*)**.** Maintain a centdian of each component in the given fully dynamic forest, where the parameter λ for each component may be different.

A naive method is to compute the centdians per update, which results in time proportional to the size of the components being updated [6]. The time complexity we would like to achieve is stated in Theorem 1.

Theorem 1. *With a linear time preprocessing, the centdian of each component can be maintained in $O(\log n)$-time, where n is the number of vertices included in the components being updated.*

Theorem 1 is proven in Section 3.3. Some properties which are used later are summarized below. Readers can refer to [6,13,15] for detailed proofs.

Property 1. *For any tree T, $v_\lambda \in P[v_c, v_m]$, and f_λ is convex along $P[v_c, v_m]$.*

Property 2. *In a rooted tree $T = (V, E)$, let $xy \in E$ and x be the parent of y. We have* $f_m(y) = f_m(x) + \left(w(\hat{T}_y) - w(T_y)\right) d(x, y)$.

Property 3. *Let xy be an edge of tree T. If x is on $P[y, v_c]$, then* $f_c(y) - f_c(x) = d(x, y)$.