Theoretical Computer Science 554 (2014) 254–262

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

A fast algorithm for data collection along a fixed track

O. Cheong [a,1], R. El Shawi [b,c], J. Gudmundsson [b,c,*,2]

[a] Korea Advanced Institute of Science and Technology, Republic of Korea
[b] University of Sydney, Australia
[c] NICTA, Sydney, Australia

ARTICLE INFO

Available online 10 December 2013

Keywords:
Wireless sensor networks
Computational geometry
Abstract Voronoi diagram

ABSTRACT

Recent research shows that significant energy saving can be achieved in wireless sensor networks (WSNs) with a mobile base station that collects data from sensor nodes via short-range communications. However, a major performance bottleneck of such WSNs is the significantly increased latency in data collection due to the low movement speed of mobile base stations. In this paper we study the problem of finding a data collection path for a mobile base station moving along a fixed track in a wireless sensor network to minimize the latency of data collection. The main contribution is an $O(mn \log n)$ expected time algorithm, where n is the number of sensors in the networks and m is the complexity of the fixed track.

1. Introduction

Wireless sensor networks (WSNs) are a well established technology for many application areas. Their main aims are to monitor physical or environmental conditions and to cooperatively pass their data through the network to a main location. Realizing the full potential of wireless sensor networks poses research challenges ranging from hardware and architectural issues, to programming languages and operating systems for sensor networks, to security concerns, to algorithms for sensor network deployment [1].

WSNs usually consist of a large number of sensor nodes, which are battery-powered tiny devices. These devices perform three basic tasks: (i) sample a physical quantity from the surrounding environment, (ii) process the acquired data, and (iii) transfer them through wireless communications to a data collection point called sink node or base station [2,3]. The traditional WSN architectures are based on the assumption that the network is dense, so that any two nodes can communicate with each other through multihop paths. As a consequence, in most cases the sensors are assumed to be static. However, recently mobility has been introduced to WSNs and it has been shown to have several advantages, such as, increased connectivity, lower cost, higher reliability and higher energy efficiency [4,5]. An overview of Wireless Sensor Networks with Mobile Elements (WSN-MEs) can be found in the comprehensive survey by Di Francesco et al. [3].

WSN-MEs have in general three main components [1]:

Regular sensor nodes are the sources of information. They perform sensing and may also forward or relay messages in the network.
Sinks (base stations) are the destinations of information. A network usually has very few sinks.
Special support nodes perform a specific task, such as acting as intermediate data collectors or mobile gateways.

* Corresponding author at: University of Sydney, Australia
E-mail address: joachim.gudmundsson@sydney.edu.au (J. Gudmundsson).

[1] Supported in part by NRF grant 2011-0016434 and in part by NRF grant 2011-0030044 (SRC-GAIA), both funded by the government of Korea.
[2] Funded by the Australian Research Council FT100100755.

0304-3975/$ – see front matter
http://dx.doi.org/10.1016/j.tcs.2013.11.037

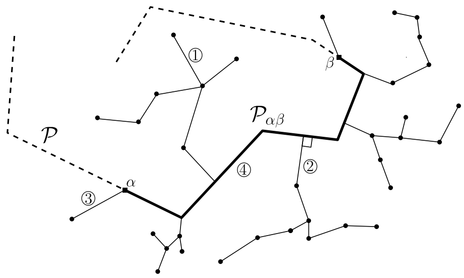

Fig. 1. Illustrating an instance of the DCFT problem with an active path $\mathcal{P}_{\alpha\beta}$ and a minimum spanning tree connecting the sensor nodes to $\mathcal{P}_{\alpha\beta}$. The encircled numbers illustrate the edge types defined in Observation 1.

In the setting considered in this paper we have one mobile sink that moves on a fixed track. The model was introduced by Xing et al. [6]. Although this is a very restricted model it simplifies the motion control of the mobile sink and it improves the system reliability and has therefore been adopted by several existing mobile sensor systems [7]. An example is when the sink only can move along fixed cables between trees [8]. The objective is to find a continuous path of length at most L along the track and a set of trees rooted on the path that connect all the sensor nodes, such that the total Euclidean length of the trees is minimized [6]. An example is shown in Fig. 1.

More formally, as input we are given a set $\mathcal{S} = \{s_1, \dots, s_n\}$ of points (sensor nodes) together with a polygonal path $\mathcal{P} = \langle p_1, \dots, p_m\rangle$ in $\mathbb{R}^2$. Given two points α and β on $\mathcal{P}$ (not necessarily vertices of $\mathcal{P}$) let $\mathcal{P}_{\alpha\beta}$ be the connected subpath of $\mathcal{P}$ with α and β as endpoints. Let $MST(\mathcal{S}, \mathcal{P}_{\alpha\beta})$ be a minimum spanning tree of $\mathcal{S}$ that contains $\mathcal{P}_{\alpha\beta}$, where an endpoint of an edge in $MST(\mathcal{S}, \mathcal{P}_{\alpha\beta})$ can either be a point of $\mathcal{S}$ or an arbitrary point on $\mathcal{P}_{\alpha\beta}$.

Problem 1 *(Data Collection on a Fixed Track (DCFT) problem).* Given a real value L, a set $\mathcal{S} = \{s_1, \dots, s_n\}$ of points and a polygonal path $\mathcal{P} = \langle p_1, \dots, p_m\rangle$ in $\mathbb{R}^2$, find a path $\mathcal{P}_{\alpha\beta} \subseteq \mathcal{P}$ of length L such that $wt(MST(\mathcal{S}, \mathcal{P}_{\alpha\beta}))$ is minimized, where $wt(\cdot)$ denotes the total length of all the edges in the tree.

The path $\mathcal{P}_{\alpha\beta}$ is called the *active* path. Since the weight of the active path is fixed the aim is to find a placement of the active path on $\mathcal{P}$ that minimizes the total weight of the trees connecting $\mathcal{S}$ to it. Note that a point in $\mathcal{S}$ can be connected to any point along $\mathcal{P}_{\alpha\beta}$, not only to the vertices and endpoints of $\mathcal{P}_{\alpha\beta}$.

To the best of the authors' knowledge very little work has been done on this problem from an algorithmic perspective and the only result we are aware of is the paper by Xing et al. [6]. They state without proof that the DCFT-problem is NP-hard, and concentrate on approximation algorithms. They present an approximation algorithm with running time $O(\frac{wt(\mathcal{P})}{\delta L} \cdot n \log n)$ and approximation ratio $\frac{2}{\sqrt{3}}(1 + 3 \cdot \frac{\delta L}{wt(\mathcal{S}, \mathcal{P}_{opt})})$, where $\mathcal{P}_{opt}$ is an optimal solution and δ is a given constant (Theorem 3 in [6]). Note that there is no relationship between L and the weight of an optimal solution, thus the approximation ratio of their algorithm can grow arbitrarily large. The same holds for the running time, the ratio between the length of $\mathcal{P}$ and L could be exponential.

In this paper we show that the DCFT-problem can in fact be solved *exactly* in $O(mn \log n)$ expected time.

2. A polynomial-time algorithm for the DCFT problem

Since the length of $\mathcal{P}_{\alpha\beta}$ is fixed, a natural approach to solve the problem is to sweep an active path of length L along $\mathcal{P}$ while maintaining a minimum spanning tree. We identify a set of $O(mn)$ event points along $\mathcal{P}$. It is shown that all topological changes to the minimum spanning tree that we maintain during the sweep occur when the start or end point of the active path coincides with one of the event points.

Two problems need to be handled: (1) find all event points along $\mathcal{P}$ of the sweep-line algorithm and (2) maintain an $MST(\mathcal{S}, \mathcal{P}_{\alpha\beta})$ during the sweep.

2.1. Basic properties and notations

Given a point s and a segment ℓ in the plane let the projection of s on ℓ, $op(s, \ell)$, be the closest point on ℓ to s, see Fig. 2. Similarly, let $op(s, \mathcal{P}) = \bigcup_{\ell \in \mathcal{P}} op(s, \ell)$ and let $op(\mathcal{S}, \mathcal{P}) = \bigcup_{s \in \mathcal{S}} op(s, \mathcal{P})$.

Next we study the edges in an optimal solution in more detail (see Fig. 1).

Observation 1. An edge $(u, v) \in MST(\mathcal{S}, \mathcal{P}_{\alpha\beta})$ can be one of four types:

Type-1: $u, v \in \mathcal{S}$,
Type-2: $u \in \mathcal{S}$ and $v \in op(u, \mathcal{P}_{\alpha\beta})$ and v lies inside $\mathcal{P}_{\alpha\beta}$,
Type-3: $u \in \mathcal{S}$ and v is either α or β, or
Type-4: u and v are consecutive vertices of $\mathcal{P}$ or points in $op(\mathcal{S}, \mathcal{P})$ along $\mathcal{P}$ (these are the edges forming $\mathcal{P}_{\alpha\beta}$).