Theoretical Computer Science 605 (2015) 51–61

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

The minimum backlog problem

Michael A. Bender[a], Sándor P. Fekete[b,*], Alexander Kröller[b], Vincenzo Liberatore[c], Joseph S.B. Mitchell[d], Valentin Polishchuk[e], Jukka Suomela[f,g]

[a] Department of Computer Science, Stony Brook University, USA
[b] Department of Computer Science, TU Braunschweig, Germany
[c] Department of Computer Science, Case Western Reserve University, USA
[d] Department of Applied Mathematics and Statistics, Stony Brook University, USA
[e] Communications and Transport Systems, ITN, Linköping University, Sweden
[f] Helsinki Institute for Information Technology HIIT, Department of Computer Science, University of Helsinki, Finland
[g] Department of Computer Science, Aalto University, Finland

ARTICLE INFO

Article history:
Received 12 November 2014
Received in revised form 17 August 2015
Accepted 24 August 2015
Available online 29 August 2015
Communicated by A. Marchetti-Spaccamela

Keywords:
Online algorithms
Competitive analysis
Computational geometry
Games on graphs

ABSTRACT

We study the minimum backlog problem (MBP). This online problem arises, e.g., in the context of sensor networks. We focus on two main variants of MBP.

The *discrete MBP* is a 2-person game played on a graph $G = (V, E)$. The *player* is initially located at a vertex of the graph. In each time step, the *adversary* pours a total of one unit of water into *cups* that are located on the vertices of the graph, arbitrarily distributing the water among the cups. The player then moves from her current vertex to an adjacent vertex and empties the cup at that vertex. The player's objective is to minimize the *backlog*, i.e., the maximum amount of water in any cup at any time.

The *geometric MBP* is a continuous-time version of the MBP: the cups are points in the two-dimensional plane, the adversary pours water continuously at a constant rate, and the player moves in the plane with unit speed. Again, the player's objective is to minimize the backlog.

We show that the *competitive ratio* of any algorithm for the MBP has a lower bound of $\Omega(D)$, where D is the diameter of the graph (for the discrete MBP) or the diameter of the point set (for the geometric MBP). Therefore we focus on determining a strategy for the player that guarantees a uniform upper bound on the absolute value of the backlog.

For the absolute value of the backlog there is a trivial lower bound of $\Omega(D)$, and the deamortization analysis of Dietz and Sleator gives an upper bound of $O(D \log N)$ for N cups. Our main result is a tight upper bound for the geometric MBP: we show that there is a strategy for the player that guarantees a backlog of $O(D)$, independently of the number of cups.

We also study a *localized* version of the discrete MBP: the adversary has a location within the graph and must act locally (filling cups) with respect to his position, just as the player acts locally (emptying cups) with respect to her position. We prove that deciding the value of this game is PSPACE-hard.

* Corresponding author.
E-mail address: s.fekete@tu-bs.de (S.P. Fekete).

http://dx.doi.org/10.1016/j.tcs.2015.08.027

1. Introduction

We study the **minimum backlog problem** (**MBP**). This is an online problem in which an agent moves around a domain and services a set of locations, "emptying" a buffer at each location in an effort to make sure that no buffer gets too full. The MBP is related to the k-server problem [8,14,15,19,20,24] (with $k = 1$), in which requests are popping up at points in a metric space, and the k servers need to minimize the distance traveled to satisfy the requests. However, in the MBP, the objective is not to minimize the traveled distance, but to minimize the *backlog*, i.e., the maximum amount of data residing in any buffer at any point in time.

1.1. Motivation

A practical motivation for the MBP arises in the context of a sensor network that, e.g., performs motion-tracking for a set of objects that move within the sensor field. Each sensor acquires data about nearby objects. The total rate of data accumulation within the network remains approximately constant, assuming a relatively fixed set of objects being monitored; however, the distribution of the data rate over the field is nonuniform and unpredictable.

If the system is used for a field study where the data is not analyzed until the end of the experiment, it may be much more energy-efficient to store the bulk of the data locally on a memory card and let someone or something gather the data by physically visiting the sensor device (or a neighborhood of the device) [10,16,17,21,25]. During the experiment, each sensor only needs to report the amount of data in its local buffer. The objective of the data gatherer is to visit the sensors in an effective order, so that no sensor's storage device is overfull.

An analogous problem arises in scheduling battery recharging/replacement in a field of wireless devices whose power consumption varies unpredictably with time and location.

1.2. Discrete MBP

We will now formalize three versions of the MBP that we study in this paper. We start with the **discrete MBP**. In this problem, we have an unweighted directed graph $G = (V, E)$ with an (initially empty) **cup** on each vertex. There is a **player**, who moves from vertex to vertex along the edges of the graph emptying cups, and there is an **adversary** (not located anywhere in particular), who refills the cups with water. We talk about filling cups with water because of historical precedent [11].

The following game is played for an indefinite (but finite) number of rounds. In each round, the two opponents do the following:

- The adversary pours a total of one unit of water into the cups. The adversary is free to distribute the unit of water in any way he likes. The adversary may base his decision on the current location of the player and the current water levels in all of the cups.
- The player moves along an edge and empties the cup in its new location. The player can see the amount of water that the adversary has poured into each cup, and the player can use this information to make the decisions.

The problem is online, i.e., the player does not know how the water is distributed in the future. Thus, the player's decision of which edge to traverse may be based only on the amount of the water that has been poured into all cups so far, but not on the future distribution of water. The objective of the player is to minimize the **backlog**, which is defined to be the maximum amount of water in any cup at any time.

1.3. Geometric MBP

It is straightforward to generalize the MBP to weighted, continuous scenarios. In this paper we will mainly focus on the following version which we call the **geometric MBP**; this version is of particular interest for sensor-network applications. Let $P \subseteq \mathbb{R}^2$ be a finite planar set. There is a cup on each point of P. The two-player game proceeds as follows in a continuous manner:

- The adversary pours water into the cups P; the total rate at which the water is poured into the cups is 1.
- The player moves in the plane with unit speed, starting at an arbitrary point. Whenever the player visits a cup, the cup is emptied.

Again, this is an online problem, and the goal of the player is to minimize the backlog.

1.4. Localized MBP

In the discrete and geometric versions of the MBP, the actions of the player are restricted by her location. To keep the game fair, we may also consider the following variant in which the actions of the adversary are also restricted by his