Theoretical Computer Science 584 (2015) 144–154

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

A nonmonotone analysis with the primal–dual approach: Online routing of virtual circuits with unknown durations

Guy Even, Moti Medina [*,1]

School of Electrical Engineering, Tel-Aviv Univ., Tel-Aviv 69978, Israel

ARTICLE INFO

Article history:
Received 5 October 2013
Received in revised form 3 September 2014
Accepted 1 October 2014
Available online 18 October 2014

Keywords:
Online algorithms
Primal–dual approach
Nonmonotone analysis

ABSTRACT

We address the question of whether the primal–dual approach for the design and analysis of online algorithms can be applied to nonmonotone problems. We provide a positive answer by presenting a primal–dual analysis to the online algorithm of Awerbuch et al. [3] for routing virtual circuits with unknown durations.

1. Introduction

The analysis of most online algorithms is based on a potential function (see, for example, [2,4,1,3] in the context of online routing). Buchbinder and Naor [6] presented a primal–dual approach for analyzing online algorithms. This approach replaces the need to find the appropriate potential function by the task of finding an appropriate linear programming formulation.

The primal–dual approach presented by Buchbinder and Naor has a monotone nature. Monotonicity means that: (1) Variables and constraints arrive in an online fashion. Once a variable or constraint appears, it is never deleted. (2) Values of variables, if updated, are only increased. We address the question of whether the primal–dual approach can be extended to analyze nonmonotone algorithms.[2]

An elegant example of nonmonotone behavior occurs in the problem of online routing of virtual circuits with unknown durations. In the problem of routing virtual circuits, we are given a graph with edge capacities. Each request r_i consists of a source-destination pair (s_i, t_i). A request r_i is served by allocating to it a path from s_i to t_i. The goal is to serve the requests while respecting the edge capacities as much as possible. In the online setting, requests arrive one-by-one. Upon arrival of a request r_i, the online algorithm must serve r_i. In the special case of unknown durations, at each time step, the adversary may introduce a new request or it may terminate an existing request. When a request terminates, it frees the path that was allocated to it, thus reducing the congestion along the edges in the path. The online algorithm has no knowledge of the future; namely, no information about future requests and no information about when existing requests will end. Nonmonotonicity is expressed in this online problem in two ways: (1) Requests terminate thus deleting the demand to

[*] Corresponding author.
E-mail addresses: guy@eng.tau.ac.il (G. Even), medinamo@eng.tau.ac.il (M. Medina).

[1] M.M. was partially funded by the Israeli Ministry of Science and Technology.

[2] The only instance we are aware of in which the primal–dual approach is applied to nonmonotone variables appears in [5]. In this instance, the change in the dual profit, in each round, is at least a constant times the change in the primal profit. In general, this property does not hold in a nonmonotone setting.

http://dx.doi.org/10.1016/j.tcs.2014.10.022
0304-3975/© 2014 Elsevier B.V. All rights reserved.

serve them. (2) The congestion of edges varies in a nonmonotone fashion; an addition of a path increases congestion, and a deletion of a path decreases congestion.

Awerbuch et al. [3] presented an online algorithm for online routing of virtual circuits when the requests have unknown durations. In fact, their algorithm resorts to rerouting to obtain a logarithmic competitive ratio for the load. Rerouting means that the path allocated to a request is not fixed and the algorithm may change this path from time to time. Hence, allowing rerouting increases the nonmonotone characteristics of the problem.

We present an analysis of the online algorithm of Awerbuch et al. [3] for online routing of virtual circuits with unknown durations. Our analysis uses the primal–dual approach, and hence we show that the primal–dual approach can be applied in nonmonotone settings.

The main technical contribution in this paper is an averaging argument for the change in the value of the primal LP as requests start and end (see Lemma 5). The primal–dual analysis of Buchbinder and Naor [6] relies on comparing, in each step, the changes in the values of the primal and dual LPs. For example, when a routing request arrives it increases the value of the dual by one, and when the request ends, the value of the dual is decreased by one. On the other hand, the nonmonotone nature of routing with reroutes, implies that we cannot guarantee that the change in the value of the primal LP when the request arrives is not greater than the change in the value of the primal LP when the request ends. Rather than bounding the worst case difference, we bound the average difference which suffices for the proof of the competitive ratio.

We show that the result can be extended to a resource allocation problem that generalizes the virtual circuits problem. In the resource allocation problem we consider, a request must be served by a subset of resources from a given collection of subsets. Note that this problem definition generalizes virtual circuits in which a request must be served by a path between the source and the destination. We also study the case in which the online algorithm employs an approximate min-weight oracle instead of a precise one.

2. Problem definition

2.1. Online routing of virtual circuits with unknown durations

Let $G = (V, E)$ denote a directed or undirected graph. Each edge e in E has a capacity $c_e \geq 1$. A routing request r_k is a 4-tuple $r_k = (s_k, d_k, a_k, b_k)$, where (i) $s_k, d_k \in V$ are the source and the destination of the kth routing request, respectively, (ii) $a_k \in \mathbb{N}$ is both the arrival time and the start time of the request, and (iii) $b_k \in \mathbb{N}$ is the departure time or end time of the request. Let Γ_k denote the set of paths in G from s_k to d_k. A request r_k is served if it is allocated a path in Γ_k.

Let $[N]$ denote the set $\{0, \ldots, N\}$. The input consists of a sequence of events $\sigma = \{\sigma_t\}_{t \in [N]}$. We assume that time is discrete, and event σ_t occurs at time t. There are two types of events: (i) An *arrival* of a request. When a request r_k arrives, we are given the source s_k and the destination d_k. Note that the arrival time a_k simply equals the current time t. (ii) A *departure* of a request. When a request r_k departs there is no need to serve it anymore (namely, the departure time b_k simply equals the current time t).

The set of active requests at time t is denoted by $Alive_t$ and is defined by

$$Alive_t \triangleq \{r_k \mid a_k \lneqq t \leq b_k\}.$$

An *allocation* is a sequence $A = \{p_k\}_k$ of paths such that p_k is a path from the source s_k to the destination d_k of request r_k. Let $paths_t(e, A)$ denote the number of requests that are routed along edge e by allocation A at time t, formally:

$$paths_t(e, A) \triangleq \left|\{p_k \in A : e \in p_k \text{ and } r_k \in Alive_t\}\right|.$$

The *load* of an edge e at time t is defined by

$$load_t(e, A) \triangleq \frac{paths_t(e, A)}{c_e}.$$

The *load* of an allocation A at time t is defined by

$$load_t(A) \triangleq \max_{e \in E} load_t(e, A).$$

The *load* of an allocation A is defined by

$$load(A) \triangleq \max_t load_t(A).$$

An algorithm computes an allocation of paths to the requests, and therefore we abuse notation and identify the algorithm with the allocation that is computed by it. Namely, ALG(σ) denotes the allocation computed by algorithm ALG for an input sequence σ.

In the online setting, the events arrive one-by-one, and no information is known about an event before its arrival. Moreover, (1) the length N of the sequence of events is unknown; the input simply stops at some point, (2) the departure