Bandwidth minimization algorithms

Aleksei V. Fishkin* René A. Sitters† Prudence W.H. Wong‡

1 Introduction

We study a bandwidth assignment problem in which each request (job) j has a given size s_j and the algorithm has to allocate a bandwidth b_j and a continuous time interval I_j such that $s_j = b_j \times |I_j|$, where $|I_j|$ denotes the duration of the interval I. Moreover, the interval I has to be within the release date and the due date specified for the job. The goal is to minimize the maximum bandwidth used at any time.

Bandwidth reservation is common in a lot of applications over computer networks, such as content distribution networks or mobile clients, which need bandwidth reservations to support hangovers for streaming video [1, 2]. Bandwidth reservations can be distinguished into *immediate reservations* which are made in a just-in-time manner and *advance reservations* which allow to reserve bandwidth before they are actually used. Immediate reservations can be considered as on-line version of the problem while advance reservations off-line version. The off-line problem can easily be shown to be NP-hard by a reduction from the 3-Partition problem. A $\log n/\log\log n$-approximation follows by randomized rounding of an LP-relaxation. The related problem in which the bandwidth of any job is one, i.e., we only need to specify the start times, has been well studied [3, 4]. No constant approximation algorithm is known for that problem. We believe that our problem does have an efficient constant factor approximation algorithm. Here, we summarize some preliminary results for the off-line and on-line variant.

Given an instance, let $B(t_1, t_2)$ be the total size of all jobs with release date and due date between t_1 and t_2. Then, $B^* = \max_{t_1 \leq t_2} B(t_1, t_2)/(t_2 - t_1)$ is a lower bound on the bandwidth needed. In general this lower bound can be arbitrarily bad.

Theorem 1 *If release dates and due dates are agreeable (i.e. $r_j \leq r_k$ implies $d_j \leq d_k$), then scheduling in order of earliest due date using bandwidth $B*$ for any job is optimal.*

If all sizes are equal then the problem remains non-trivial as Figure1 illustrates. In this case, scheduling in earliest due date order is 2-competitive if we use twice the bandwidth of the lower bound.

Theorem 2 *If all size are equal then scheduling in order of earliest due date using bandwidth $2B*$ for any job gives a 2-approximation.*

*avf*csc.liv.ac.uk. Department of Computer Science, University of Liverpool, Liverpool, UK.
†r.sitters*tue.nl Department of Math. and Comp. Science, TU Eindhoven, Eindhoven, The Netherlands.
‡pwong*csc.liv.ac.uk. Department of Computer Science, University of Liverpool, Liverpool, UK.

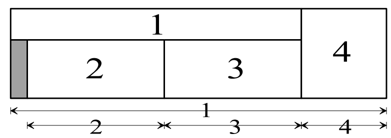

Figure 1: Even for equal sized jobs the optimal schedule might be non-trivial. The arrows indicate the release and due dates. Only job 4 uses maximum bandwidth. Job 1 does not use its entire interval. No other optimal schedule exists.

2 On-line bandwidth minimization

We assume that jobs arrive over time and jobs need to be scheduled on-line without preemption.

Theorem 3 *No deterministic on-line algorithm is better than $\frac{n}{2}$-competitive for the bandwidth minimization problem.*

If release dates and due dates are agreeable, then the following algorithm, that we call Reserve-x is constant competitive. At any moment t, let $B_t(u)$ be the total size of all jobs that are due before time $u \geq t$ and were released, but not started, before time t. Let $B_t^* = \max_{u:u>t} B_t(u)/(u-t)$. At any moment t that no job is processed, start the job with earliest due date and allocate a bandwidth of x times B_t^*.

Theorem 4 *If release dates and due dates are agreeable, then* Reserve-x *gives a feasible solution for any $x > 1$. In particular, the solution is at most 4 times optimal for $x = 2$.*

On the other hand, we can show that for agreeable jobs no on-line algorithm can be better than 2-competitive. If all jobs are of equal size, then a slightly different algorithm than above can be proven to be 8-competitive.

Theorem 5 *If all jobs have the same size, then there exists an 8-competitive algorithm.*

References

[1] L.-O. Burchard and H.-U. Heiss and C.A.F. De Rose 2003. *Performance issues of bandwidth reservations for grid computing.* Proc. 15th Symp. on Computer Architecture and High Performance Computing.

[2] D. Ferrari and A. Gupta and G. Ventre 1995 *Distributed advance reservation of real-time connections.* Proc. 5th. Int. Workshop on Network and Operating Systems Support for Digital Audio and Video.

[3] J. Chuzhoy and S. Guha and S. Khanna and J. Naor 2004. *Machine minimization for scheduling jobs with interval constraints.* Proc. 45th. Symposium on Foundations of Computer Science.

[4] J. Chuzhoy and J. Naor 2004. *New hardness results congestion minimization and machine scheduling.* Proc. 36th. Annual ACM Symposium on Theory of Compting.