

Erratum for “Meeting deadlines: How much speed suffices?”

S. Anand* Naveen Garg† Nicole Megow†

March 30, 2015

This is an erratum for “Meeting deadlines: How much speed suffices?” which appeared in the Proceedings of ICALP 2011 [1]. There is an incorrect claim in the analysis of the algorithm presented in the paper, which invalidates the main result of the paper.

The false claim appears on Page 7 in the proof that the speed $\alpha_m = 1/(1 - (1 - \frac{1}{m})^m) < e/(e - 1) \approx 1.58$ is sufficient to schedule on m machines all the work volume that the algorithm assigned to feasible time slots. It is claimed that one may assume that the worst case happens when all jobs arrive at the same time, that is, $t = 0$. The following counter example, found by Kevin Schewior, shows that this is wrong.

Example 1. *The instance consists of four jobs with deadlines $d_j = j$, $j \in \{1, 2, 3, 4\}$. Jobs 1, 2 and 4 are released at time 0, whereas job 3 is released at time 1. The processing times are as follows: $p_j = j$, $j \in \{1, 2, 4\}$, and $p_3 = 1$. There are $m = 2$ machines.*

Following the algorithm in [1], job 4 fails to be feasibly scheduled in time slot $(1, 2]$ with a speed of $\alpha_2 = 4/3$. In the schedule constructed before the release of job 3, job 4 is assigned with a processing volume $1/3$ to interval $(0, 1]$, with $4/3$ to each of the slots $(1, 2]$ and $(2, 3]$ and with one unit of processing to $(3, 4]$. At time 1 after incorporating the new job 3, job 3 is assigned with a processing volume of $4/3$ to each of the slots $(2, 3]$ and $(3, 4]$. Since only $1/3$ has been finished in $(0, 1]$ there remains one unit of processing to be done in $(1, 2]$. However, in this time slot are already one unit for each of the two not-underworked jobs 2 and 3. The total volume of 3 cannot be feasibly scheduled in a unit time slot when given speed $4/3$.

This simple example can be extended by additional jobs such that, for $m = 2$, the required speed is at least $2 - 1/m = 3/2$. For $m = 3$, a similar construction gives a lower bound of $5/3 - \varepsilon$, for any $\varepsilon > 0$. We believe that the true speedup bound for the algorithm is $2 - 1/m$ for any number of machines m , but currently we cannot give a proof. The currently best known online algorithm for the online deadline scheduling problem requires a speed factor of $2 - 2/(m + 1)$ [2].

We remark that the lower bound results in Sections 4 and 5 remain correct.

Acknowledgement. We thank Kevin Schewior for finding the counter examples for the claim.

References

- [1] S. Anand, N. Garg, and N. Megow. Meeting deadlines: How much speed suffices? In *Proceedings of the 38th International Colloquium on Automata, Languages and Programming (ICALP 2011)*, volume 6755 of *LNCS*, pages 232–243. Springer, 2011.
- [2] T. W. Lam and K.-K. To. Trade-offs between speed and processor in hard-deadline scheduling. In *Proc. of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 623–632, 1999.

*Indian Institute of Technology Delhi, India. Email: anand.42@gmail.com, naveen@cse.iitd.ac.in

†Technische Universität Berlin, Germany. Email: nmegow@math.tu-berlin.de