Resource Augmentation Analysis of Scheduling Problems

Kirk R. Pruhs

Computer Science Department, University of Pittsburgh, kirk@cs.pitt.edu

We consider settings where the scheduler can not produce optimum schedules, with the most common reason being that either the scheduling problem is NP-hard and the scheduler is computationally bounded, or the scheduler must schedule tasks online as they arrive, as is the case in an operating system, web server, etc. The most common way to measure the Quality of Service (QoS) of a scheduling algorithm in these settings is to use the worst case ratio of the QoS of the computed schedule to the QoS of the optimal scheduling. Unfortunately, for many problems this ratio can be unbounded for all possible algorithms. If one examines the troublesome instances, one generally finds that they fully load the scheduler. This makes intuitive sense in that if the scheduler makes a mistake when it is fully loaded then it does not have any spare resources to correct for even the smallest error. Resource augmentation analysis compares the quality of the computed schedule given slightly more resources (e.g. a faster processor) against the optimal schedule with less resources. In some sense increasing the resources corresponds to lowering the load. Thus resource augmentation results can intuitively show that a scheduling policy is good at loads below peak server capacity. Resources augmentation analysis often gives strikingly better results than standard worst case analysis, and also for many problems identifies the algorithms that have proven to be the experimental champions, e.g. Greedy-Dual-Size for browser caching and MLF for CPU process scheduling. In this talk I will introduce resource augmentation analysis, give some examples of the type of results that one obtains, and then give a few outstanding open questions.

The recent popularity of resource augmentation analysis of scheduling problems emanates from [7]. The term *resource augmentation*, and the associated terminology is from [9]. I will discuss resource augmentation analysis within the context of three problems:

- Nonclairvoyant Scheduling to Minimize Total Flow Time: The algorithm Shortest Elapsed Time First (SETF), which always runs the job that has been run the least, has bounded error if it has a $(1+\epsilon)$-speed processor[7].
- Scheduling Jobs with Arbitrary Speed-up Curves on Parallel Processors: Round Robin has bounded error if it has a $(2+\epsilon)$-speed processor[1].
- Multicast Pull Scheduling: Offline constant-speed constant-approximation LP-based polynomial-time algorithms are given in [4, 8, 5, 6]. Online constant-speed constant-approximation algorithms are given in [2, 3].

Since the publication of [7] resource augmentation has been applied to a wide array of scheduling problems. For a survey see [10].

Acknowledgments: Supported in part by NSF grant CCR-0098752, NSF grant ANIR-0123705, NSF grant ANI-0325353, and a grant from the United States Air Force.

References

[1] J. Edmonds, "Scheduling in the dark", *Theoretical Computer Science*, **235**(1), 109 – 141, 2000.

[2] J. Edmonds and K. Pruhs, "Broadcast scheduling: when fairness is fine", Symposium on Discrete Algorithms (SODA) 2002.

[3] J. Edmonds and K. Pruhs, "A maiden analysis of Longest Wait First", Symposium on Discrete Algorithms (SODA) 2004.

[4] T. Erlebach and A. Hall, "Hardness of broadcast scheduling and inapproximability of single-source unsplittable min-cost flow", Symposium on Discrete Algorithms (SODA) 2002.

[5] R. Gandhi, S. Khuller, Y. Kim and Y-C. Wan, "Approximation algorithms for broadcast scheduling", Conference on Integer Programming and Optimization (IPCO), 425- 438, 2002.

[6] R. Gandhi, S. Khuller, S. Parthasarathy, A. Srinivasan, "Dependent rounding in bipartite graphs," IEEE Symposium on Foundations of Computer Science (FOCS), 2002

[7] B. Kalyanasundaram, and K. Pruhs, "Speed is as powerful as clairvoyance", *Journal of the ACM*, **47**(4), 617 – 643, 2000.

[8] B. Kalyanasundaram, K. Pruhs, and M. Velauthapillai, "Scheduling broadcasts in wireless networks", *Journal of Scheduling*, **4**(6), 339 – 354, 2000.

[9] C. Phillips, C. Stein, E. Torng, and J. Wein "Optimal time-critical scheduling via resource augmentation", *Algorithmica*, **32**(2), 163 – 200, 2002.

[10] K. Pruhs, J. Sgall, and E. Torng, "Online Scheduling", to appear in *Handbook on Scheduling: Algorithms, Mod els and Performance Analysis*, CRC press. Temporarily at `http://www.cs.pitt.edu/~kirk/papers/onlineschedulingsurvey.ps`.