Proposal of Research

Jessica Chang

Scheduling problems arise in any circumstance where information must be distributed in the presence of constrained resources. While specific variations of the scheduling problem have been extensively studied, other newer models have emerged as technological development has changed the way that information is disseminated.

Broadcast Scheduling

One of the most relevant scheduling models today is that of broadcasting, in which a server can broadcast requested information (or pages) to satisfy multiple demands for these data. The time that a client has to wait for his requests to be satisfied is called his *response time*. Where the scheduling problem is NP-hard (which is the case for several interesting scheduling problems), efforts have been devoted to designing approximation algorithms and analyzing their performance bounds. While broadcast scheduling has received significant attention in recent years, there still remain several open problems of particular interest.

With respect to minimizing the maximum response time, I proved a lower bound of 2 for the online problem in [1]. However, there is no reason to believe that an approximation factor of less than 2 cannot be achieved in the offline case[1]; in fact, no lower bounds are known.

One can approach this model from another perspective; suppose that the server is given a fixed maximum response time T and must try to satisfy as many requests as possible given that no request can experience delay greater than T. The best approximation algorithm uses a technique known as LP-rounding. This algorithm guarantees to satisfy at least $\frac{3}{4}$ of the number of requests that the optimal solution satisfies [5]. We know from [1] that we cannot do better than $\frac{17}{18}$; can this bound be improved? Can other approaches approximate the solution better? Also, the formulated program used in the LP-rounding technique requires knowledge of all requests prior to the development of the schedule. What methods are ideal for approximating this problem in the online setting?

I am also interested in another variant of broadcast scheduling, which is described as follows. Consider the scenario where the server is allowed to drop a small percentage of requests. Then in the optimal schedule, the maximum response time over all serviced requests may be considerably smaller than that of the schedule which must service all requests. How close can we get to optimal in this scenario? The current approximation factor is 5 [2]. It would be interesting to understand how this bound can be improved.

These various broadcasting models arise because there are scenarios in which satisfying all jobs is not more important than reducing the response time for most requests. The minimization of maximum response time is of particular interest to the client, who may not care about the general performance of the schedule, but only about the time that he will have to wait for his request to be satisfied. Interestingly enough, while these variants are related in some sense, their analyses and approximations may be dissimilar.

[1] An algorithm is a α-approximation means that it performs no worse than α times the optimal solution.

Mechanism Design in Scheduling

I am also becoming increasingly interested algorithmic mechanism design, particularly as it applies to scheduling scenarios. In [6], Nisan and Ronen considered the task scheduling problem on a distributed system where resource providers (machines) are not necessarily inclined to follow a particular protocol unless it aligns with their self-interest.

In particular, we have n machines to process a group of m different tasks, but the time required for a machine to process a particular task varies from machine to machine and is known only by that machine. Each machine has the selfish goal of minimizing the amount of work processed on it, while the goal of the mechanism is to minimize the completion time of the last task (i.e. makespan). This is how "social welfare" is quantified in this model. Because of these differing goals, we want to design the mechanism in a way that motivates the self-interested machine to reveal his true times (as opposed to, for example, overstating his processing times so that the mechanism will allocate tasks to a different machine). The standard approach is to introduce the notion of money, i.e. payments that can offset a machine's preferences. Part of designing a mechanism involves determining what these payments should be. The best known mechanism for doing this achieves an approximation factor of n, and this bound tight for certain cases. However, in the general case, the lower bound is provably $1 + \sqrt{2}$ for $n \geq 3$ [3], but is conjectured to be much higher [6]. This problem is still very open, and it would be very interesting to see how this large gap can be improved.

Theoretical computer science is a big and expanding field; there remains much to learn and much to be discovered. My general interest involves the study of algorithms, particularly its design and analysis.

References

[1] J. Chang, T. Erlebach, R. Gailis and S. Khuller. Broadcast Scheduling: Algorithms and Complexity. To appear in *Proceedings of 19th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2008.

[2] M. Charikar and S. Khuller. A Robust Maximum Completion Time Measure for Scheduling. *Proceedings of 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2006.

[3] G. Christodoulou, E. Koutsoupias and A. Vidali. A lower bound for scheduling mechanisms. *Proceedings of 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2007.

[4] R. Gandhi, S. Khuller, S. Parthasarathy and A. Srinivasan. Dependent Rounding in Bipartite Graphs. *Proceedings of IEEE Symposium on Foundations of Computer Science*, 2002.

[5] N. Nisan and A. Ronen. Algorithmic Mechanism Design. *Proceedings of 31st Annual ACM Symposium on Theory of Computing*, 1999.