Efficient Allocations when the Agents have Submodular Utility/Cost Function

Gagan Goel

Given a set of agents A, a set of indivisible items I, and for each agent $i \in A$, a function $f_i : 2^I \rightarrow R^+$, the goal is to allocate items to agents with certain objective in mind. Function f_i can be interpreted as a utility function or a cost function depending on what the items correspond to. If the items correspond to goods that bring happiness to the agents, then the function f_i is interpreted as a utility function. On the other hand if the items correspond to things like projects which have inherent cost of doing it, then the function f_i is interpreted as a cost function. In both the cases, a natural assumption on f_i is that it is submodular.

Depending on a particular application, some of the objective functions that are typically of interest are:

- *Maximize* the total value i.e. $max \sum_i f_i(S_i)$
- *Maximize the minimum* value i.e. $max\ min_i(f_i(S_i))$
- *Minimize* the total cost i.e. $min \sum_i f_i(S_i)$
- *Minimize the maximum* cost i.e. $min\ max_i(f_i(S_i))$

I am interested in working on the following *three algorithmic questions* which fall in the above broad setting:

A) Maximize the total value when the submodular function is of special type - linear with a budget constraint. Motivation of this problem comes from the allocation issues in first price budgeted auctions. Recently I (along with Deeparnab Chakrabarty) gave a best known approximation factor of 3/4 and hardness of 15/16. I am interested in further reducing this complexity gap.

B) Minimize the total cost for general monotone submodular function. Recently I showed how to get a ln(n) factor algorithm (with a matching NP hardness) for this problem. Further questions that I intend to work on are: 1) Unconditional communication complexity based lower bounds 2) Truthful auction design for this setting.

C) Minimize the maximum cost when the submodular function correspond to set cover type of functions. This routine is used at search engines like Google to store the web documents on different servers with the goal of minimizing the maximum load. Siva kumar(research scientist at Google) told me about this problem, and said that the greedy type heuristics used in practice perform badly. Recently, Fleischer et. al. gave ($\sqrt{|I|}$) approximation algorithm for this problem when the functions are any monotone submodular function. I intend to work on this special type of monotone submodular function whose motivation comes from the real world application of storing web documents. Our initial study suggests that the better guarantees are possible for this special case.