CSC2420 - Algorithm Design, Analysis and Theory **Fall 2022**

Lecture 7: Local Search and Randomization

Lecturer: Allan Borodin *Scribe: Kevin Gao*

7.1 Facility Location

Problem 7.1 ((Uncapacitated) Metric Facility Location Problem)**.**

Input: (F, C, d, f) where F is a set of facilities, C is a set of clients/cities, d is a metric distance functoin over $F \cup C$, and f is an opening cost function for facilities.

Output: A subset of facilities F' minimizing $\sum_{i \in F'} f_i + \sum_{j \in C} d(j, F')$ where f_i is the opening cost of facility $i \in F$ and $d(j, F') = \min_{i \in F'} d(j, i)$.

In the capacitated version, facilities have capacities and cities can have demands (rather than unit demand). The constraint is that a facility cannot have more assigned demand than its capacity so that it is not possible to always assign a city to its closest facility.

Problem 7.2 (k-Median Problem)**.**

Input: (F, C, d, k) where F, C, d are as in UFL and k is the number of facilities that can be opened.

Output: A subset of facilities F' with $|F'| = k$ that minimizes $\sum_{j \in C} d(j, F')$.

UFL is hard to approximate to within a factor better than 1.463 assuming P $\not\subseteq$ DTIME$(n^{\log \log n})$ (Guha, Khuller 1999). the k-Median problem is hard to approximate to within a factor of $1 + 1/e \approx 1.736$ (Jain, Mahdian, Saberi). The current best polynomial time approximation for UFL is 1.488 (Li, 2011).

7.2 k-independence Systems

There are many ways to extend matroids. In particular, the exchange property immediately implies that in a matroid M, every maximal independent set (called a base) has the same cardinality, the rank of M. Matroids are those independence systems where all bases have the same cardinality. Let k be a positive integer. A (Jenkyns) k-independence system satisfies the weaker property that for any set S and two bases B and B_0 of S, $|B|/|B_0| \leq k$. When $k = 1$, the system is a matroid. An example of a k-independent system is the intersection of k matroids.

7.3 Submodular Functions

Let U be a universe. We consider the set of functions $f(S) \geq 0$ for all $S \subseteq U$. We will also assume that the functions are normalized in that $F(\emptyset) = 0$.

A **sublinear** set function satisfies the property that

$$f(S \cup T) \leq f(S) + f(T)$$

for all subsets $S, T \subseteq U$.

When $f(S \cup T) + f(S \cap T) = f(S) + f(T)$, the function is a **linear** (or **modular**) function.

A **submodular** set function $f : U \to \mathbb{R}$ satisfies the property that

$$f(S \cup T) + f(S \cap T) \leq f(S) + f(T)$$

It follows that modular set functions are submodular and submodular functions sublinear. Submodular functions can be monotone or non-monotone. A monotone submodular function also satisfies the property that

$$f(S) \leq f(T)$$

whenever $S \subseteq T$.

Submodular function can be equivalently characterized as those satisfying the **decreasing marginal gain** property, where for $S \subset T$

$$f(T \cup \{x\}) - f(T) \leq f(S \cup \{x\}) - f(S)$$

That is, adding additional elements has non-increasing marginal gain for larger sets. In other words, we gain more by adding elements to a smaller set than to a larger set. This is also called **diminishing returns**: the more you have, the less you want.

Some observations:

- Modular functions are monotone.
- The rank of a matroid is a monotone submodular function.
- The two most common examples of non-monotone submodular functions are max-cut and max-di-cut.

The monotone problem is only interesting when the submodular maximization is subject to some constraint. One of the simplest constraints is a cardinality constraint: to maximize $f(S)$ subject to $|S| \leq k$ for some k; since f is monotone, this is equivalent to requiring $f(S) = k$. The following greedy algorithm for the constrained monotone function maximization problem with the approximation ratio $1 - (1 - 1/k)^k$ approximation for a k cardinality constraint.

$S = \emptyset$
while $|S| \leq k$
    $u = \arg\max_u f(S \cup \{u\}) - f(S)$
    $S = S \cup \{u\}$

Theorem 7.3. *The above algorithm achieves an approximation ratio of* $1 - (1 - 1/k)^k$ *for monotone submodular function maximization subject to cardinality constraint.*

Proof:

■

7.4 Randomized Algorithms

There are some problem settings (e.g. simulation, cryptography, interactive proofs, sublinear algorithms) where randomization is necessary. We can also use randomization to improve approximation ratios.

In complexity theory, a fundamental question is how much can randomization lower the time complexity of a problem. For decision problems, there are three polynomial time randomized classes ZPP (zero-sided), RP (one-sided), and BPP (two-sided) error. The big question in **fine-grained complexity theory** is BPP $\stackrel{?}{=}$ P.

One important aspect of randomized algorithms in an offline setting is that the probability of success can be amplified by repeated independent trials of the algorithm.