18.1 Local Search and k-Median

Local search is a general technique and it is the underlying idea for hill-climbing, steepest descent and simulated annealing algorithms. The general idea to local search is that the algorithm searches for a local optimum, and then hope that this optimum is close to the global optimum.

Let Π be an optimization problem (in this case a minimization problem a similar definition can be given for maximization problems) and I be an instance of the problem. Suppose $S_\Pi(I)$ is the set of all feasible solutions, also known as the solution space. For any solution $s \in S_\Pi(I)$, $c_\Pi(s, I)$ is the cost of s. We are searching for a solution that minimizes c_Π. Let $n_\Pi(s, I)$ be the neighborhood of solution s which is obtained by changing solution s by some defined rule. For example, if a solution s is a $\{0, 1\}$ vector, solutions within a certain Hamming distance of s could be considered to be the neighborhood. Different rules define different neighborhoods.

General Steps in Local Search Algorithms

Pick some feasible solution $s \in S_\Pi(I)$
Repeat:
  Find $s' \in n_\Pi(s, I)$ such that $c_\Pi(s', I) < c_\Pi(s, I)$
  If s' is found then $s \leftarrow s'$
Until s' is not found
Return s

An initial solution may be chosen by various methods such as random selection or greedy search. The solution returned by this general algorithm will be a local optimum.

Definition 18.1 *Locality Gap*: The *Locality Gap* for a given neighborhood n_Π is defined as:

$$\max_I \frac{\text{Local Optimum}(I)}{\text{Global Optimum}(I)}$$

The locality gap is the worst case ratio between a local optimum and a global optimum. Proving a good upper bound for this gap is crucial in computing the approximation ratio of algorithms.

Another important consideration with the general algorithm is the number of iterations of the while loop (the algorithm's running time).

18.2 k-Median

Is a variation of UFL, where facilities don't have openning cost but we have a limit on the number of facilities we can open.

Input:

- F, a set of facilities ($|F| = n$);
- C, a set of cities/clients/users, ($|C| = m$);
- For all $1 \leq i \leq n, 1 \leq j \leq m$: C_{ij} is the cost of connecting city j to facility i;
- k, the maximum number of facilities that we can open (there is no opening cost)

Goal:

- Find a subset $S \subseteq F$, with $|S| \leq k$, to be openned, and connect each city to an open facility such that the total connection cost is minimized.

Remark: We are in a disjoint (bipartite) setting (i.e. $F \cap C = \emptyset$). In the *complete* setting, $F = C$. However, the complete case can be reduced to the bipartite case. We also assume that we are in the metric case which satisfies the triangle inequality.

Observation 18.2 *We can assume that the optimal solution has k open facilities. This holds because there is no opening cost on facilities, so adding facilities that are not connected to any city will not have any impact on the cost of the solution.*

Observation 18.3 *Once we fix the set of facilities, each city must be connected to the nearest facility (for every city j, the facility i such that c_{ij} is minimized). Thus finding the set of k facilities to open determines the connections as well.*

Any solution is a $\{0,1\}^n$ vector with exactly k ones, where each one corresponds to an open facility. Our neighborhood for a solution S will be the vectors with Hamming distance 2 from S, i.e. they differ by a single *swap* operation. A *swap* $< s, s' >$ with $s \in S$ and $s' \notin S$ yields $S - s + s'$.

k-Median Local Search Algorithm

$S \leftarrow$ an arbitrary set of k facilities
while there is a swap operation *op* with $cost(op(S)) < (1 - \frac{\epsilon}{P(n,m)})cost(S)$ do:
    $S \leftarrow op(S)$
return S

Here $P(n,m)$ is some polynomial in terms of n and m and ϵ is an arbitrarily small constant. The returned solution will be within $(1+\epsilon)$ of the local optimum and within $(1+\epsilon)\alpha$ of the global optimum, for an α that we show is 5.

Trivially, we can show that there are a polynomial number of swaps possible on each iteration $(k(n-k))$. At every iteration, the cost goes down by a factor of at least $(1 - \frac{\epsilon}{P(n,m)})$. If we assume that O is an optimal solution and that s_0 is our initial solution, the number of iterations is at most:

$$\frac{\log \frac{\text{cost}(s_0)}{\text{cost}(O)}}{\log \frac{1}{1-\frac{\epsilon}{P(n,m)}}}$$

which is polynomial (for a fixed ϵ) in terms of the size of the input. Thus the number of iterations is some polynomial.

The algorithm terminates when $cost(S-s+s') \geq (1-\frac{\epsilon}{P(n,m)})cost(S)$ for any $s \in S$ and $s' \notin S$ (for any swap $< s, s' >$). To simply calculations, assume that we work with the assumption that the algorithm terminates when $cost(S-s+s') \geq cost(S)$ for any swap $< s, s' >$. Then we come back to the real condition in the algorithm and show that working with that condition doesn't change the approximation ratio by more than $(1+\epsilon')$ for an arbitrary small ϵ' that only depends on ϵ.

Here are several definitions that will be used in the analysis of the algorithm. We assume S is the result of the algorithm and that O is an optimal solution For each city j, s_j and o_j are the facilities connected/serving city j in S and O, respectively. $N_S(s)$ is the neighborhood of s in S; the set of cities connected to facility s in S. Similarly, $N_O(o)$ is the neighborhood of o in O; the set of cities connected to facility o in O. For:

$$A \subseteq S, N_S(A) = \bigcup_{s \in A} N_S(s)$$

$$B \subseteq O, N_O(B) = \bigcup_{o \in B} N_O(o)$$

Our goal is to show that:

Theorem 18.4 *The locality gap of the k-Median local search algorithm is at most 5.*

Proof: By the condition of the while loop, for any swap $< s, o >$ where $s \in S$ and $o \in O$, $cost(S-s+o) \geq cost(S)$. This is true even if $S \cap O \neq \emptyset$. We are going to show that $cost(S) \leq 5 \cdot cost(O)$.

We partition $N_O(o)$ (for every $o \in O$) into subsets $N_s^o = N_O(o) \cap N_S(s)$ for each $s \in S$. A facility $s \in S$ "captures" a facility $o \in O$ if it is connected to more than half of the cities of $N_O(o)$ in solution S. (i.e. $|N_s^o| > (1/2)|N_O(o)|$)

Observation 18.5 *Every $o \in O$ can be captured by at most 1 facility $s \in S$.*

We say $s \in S$ is 'bad' if it captures some facility $o \in O$, otherwise it is 'good'.

The proof will be concluded in Lecture 19.

AGKMMP04 Arga; Garg; Khandekar; Meyerson; Munagala; Pandit, Local Search Heuristics for k-Median and Facility Location Problems, *SIAM Journal of Computing*, 33(3), 544-562, 2003. Also appeared in STOC '01.