CMPUT 675: Approximation Algorithms **Winter 2005**

Lecture 19 : March 23

Lecturer: Mohammad R. Salavatipour *Scribe: Xiaomeng WU*

19.1 Analysis of k-Median Local Search Algorithm

Recall the k-Median problem defined last lecture:

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.

The algorithm presented was a local search algorithm. We can think of any solution as a $\{0,1\}^n$ vector with exactly k ones. The neighborhood for a solution S will be the vectors with Hamming distance 2 from S, i.e. they differ by a single *swap* operation. where a *swap* $< s, s' >$ with $s \in S$ and $s' \notin S$ yields $S - s + s'$. The algorithm is:

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.

By the condition of the while loop, we know that when the algorithm stops, the solution is not larger than the local optimum by a factor larger than $1/(1 - \frac{\epsilon}{P(n,m)}) = 1 + \epsilon'$ for some ϵ' which depends on ϵ and $p(n,m)$. Therefore, if we show that the gap between the local optimum and global optimum is at most α then the gap between our solution and the global optimum is at most $(1+\epsilon')\alpha$. Thus, to show the gap between local optimum and global optimum, we assume that the condition of the while loop has changed to "do a swap if $cost(op(S)) < cost(S)$. So the algorithm stops when no swap improves the solution by any positive amount.

Let S be the solution returned by this new local search and let O be an optimum solution. From the local optimality of S, we know that

$$cost(S - s + o) \geq cost(S) \qquad \text{for all } s \in S, o \in O. \tag{19.1}$$

Note that even if $S \cap O \neq \emptyset$, the above inequalities hold.

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:

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

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

We say $s \in S$ is 'bad' if it captures some facility $o \in O$, otherwise it is 'good'. Now we have the following claim:

Claim 19.1 *Consider a facility $o \in O$, there is a $1-1$ and onto mapping π(Figure 19.1): $N_O(o) \to N_O(o)$ that satisfies the following property: for any $s \in S$, if s does not capture o, then $\pi(N_s^o) \cap N_s^o = \emptyset$.*

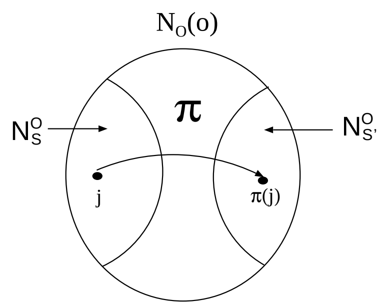

Figure 19.1: The mapping π on $N_O(o)$. s does not capture o. $s' \neq s$. π is a $1-1$ onto mapping.

One such mapping π can be constructed as follows. Order the clients in $N_O(o)$ as $c_0, \ldots, c_{|N_O(o)|-1}$ such that for every $s \in S$ with a nonempty N_s^o, the clients in N_o^s are consecutive. Define $\pi(c_j) = c_j$, where $j = (i + \lfloor |N_O(o)|/2 \rfloor) \ mod \ (|N_O(o)|)$. That is, we map everything one half ahead.

To see such a mapping satisfies the above property, assume both c_j, $\pi(c_j) = c_j \in N_s^o$ for some s, where $|N_s^o| \leq |N_O(o)|/2$. If $j = i + \lfloor |N_O(o)|/2 \rfloor$, then $|N_s^o| \geq j - i + 1 = \lfloor |N_O(o)|/2 \rfloor + 1 > \lfloor |N_O(o)|/2 \rfloor$. If $j = i + \lfloor |N_O(o)|/2 \rfloor - |N_O(o)|$, then $|N_s^o| \geq i - j + 1 = |N_O(o)| - \lfloor |N_O(o)|/2 \rfloor + 1 > \lfloor |N_O(o)|/2 \rfloor$. In both cases, we have a contradiction.

Based on the notion of *"capture"*, we can construct a bipartite graph $H(S, O, E)$ (Figure 19.2) in this way: for each facility in S, there is a vertex on the S-side, and for each facility in O, there is a vertex on the O-side. An edge $s_i o_j \in E$ iff s_i captures o_j. H is called the *capture graph*, which has this property: each vertex in O has degree at most 1, and vertices in S have degrees up to k (that is, 0, or ≥ 1).

We now consider k swaps (one for each facility in O). If some bad facility $s \in S$ captures exactly one $o \in O$, then we consider the swap $\langle s, o \rangle$.

Suppose l facilities in S (and so l vertices in O) are not considered in such swaps, there must be $\geq l/2$ good facilities in S; because each facility out of these l facilities in S is either good or captures at least two

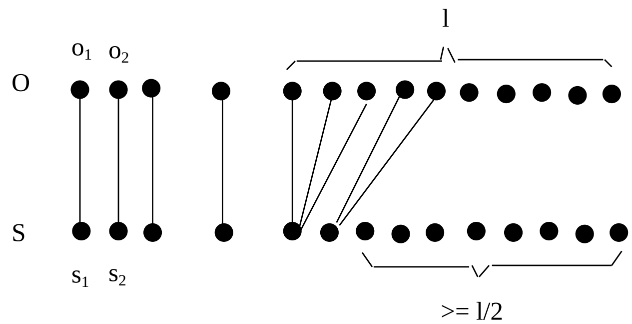

Figure 19.2: Capture graph $H(S, O, E)$.

facilities in O. Now, consider l swaps in which the remaining l facilities in O get swapped with the good facilities in S such that each good facility is considered in at most two swaps (Figure 19.3). The bad swaps which capture at least two facilities in O are not considered in any swaps.

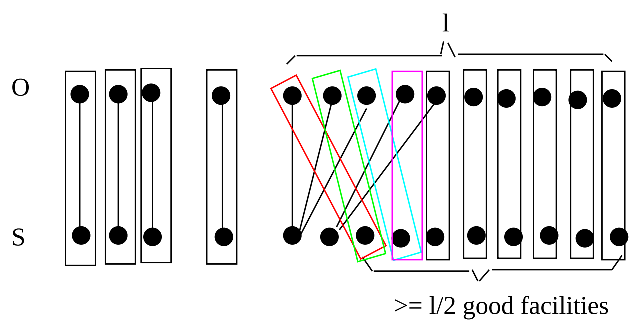

Figure 19.3: k swaps considered in the analysis

These k swaps satisfy the following properties:

1. Each $o \in O$ is considered in exactly one swap.
2. Every facility that captures more than two is not in any swap.
3. Each good facility $s \in S$ is considered in at most two swaps.
4. If $\langle s, o \rangle$ is considered, then facility s does not capture any facility $o' \neq o$.

Consider one of these k swaps $\langle s, o \rangle$, we will show an upper bound on the increase in the cost by re-assigning the cities in $N_O(o) \cup N_S(s)$ to the facilities in $S - s + o$ as follows (Figure 19.4):

- (a) Every city $j \in N_O(o)$ is now assigned to o.
- (b) All the cities not in $N_S(s) \cup N_O(o)$ continue to be served by the same facility.

Consider a city $j' \in N_s^{o'}$ for $o' \neq o$. As s does not capture o', by the claim about mapping π, we have that $\pi(j') \notin N_S(s)$. Let $\pi(j') \in N_S(s')$. Note that the distance the city j' to the nearest facility in $S - s + o$ is at most $c_{j's'}$. From triangle inequality, we have $c_{j's'} \leq c_{j'o'} + c_{\pi(j')o'} + c_{\pi(j')s'} = O_{j'} + O_{\pi(j')} + S_{\pi(j')}$.

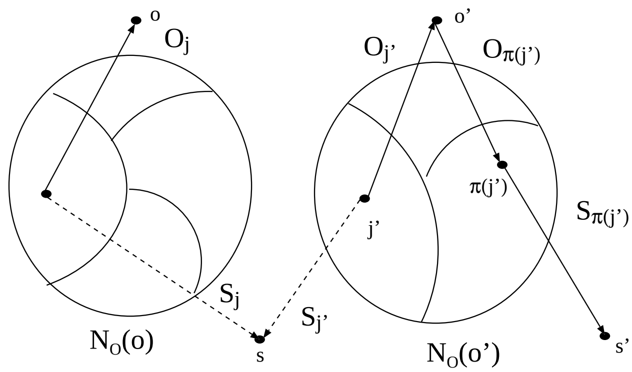

Figure 19.4: Reassigning the cities in $N_S(s) \cup N_O(o)$

Combining the inequality $cost(S - s + o) - cost(S) \geq 0$, we have the summation:

$$\sum_{j \in N_O(o)} (O - S) + \sum_{j \in N_S(s), j \notin N_O(o)} (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j) \geq 0 \quad (19.2)$$

As each facility $o \in O$ is considered in exactly one swap, the first term of above inequality added over all k swaps gives exactly $cost(O) - cost(S)$. For the second term, we will use the fact that each $s \in S$ is considered in at most two swaps. Since S_j is the shortest distance from city j to a facility in S, using triangle inequality we get $O_j + O_{\pi(j)} + S_{\pi(j)} \geq S_j$. Thus the second term added over all k swaps is no greater than $2\sum_{j\in C}(O_j + O_{\pi(j)} + S_{\pi(j)} - S_j)$. As π is a $1-1$ and onto mapping, $\sum_{j\in C} O_j = \sum_{j \in C} O_{\pi(j)} = cost(O)$ and $\sum_{j\in C}(S_{\pi(j)} - S_j) = 0$. Thus, $2\sum_{j\in C}(O_j + O_{\pi(j)} + S_{\pi(j)} - S_j) = 4cost(O)$. Combining the two terms, we get: $cost(O) - cost(S) + 4cost(O) \geq 0$, that is, $cost(S) \leq 5cost(O)$.

If p facilities can be swapped simultaneously instead of one swap, the locality gap is $3 + 2/p$. The details can be found in [AGKMMP04].

References

AGKMMP04 V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit Local Search Heuristics for k-Median and Facility Location Problmes. *SIAM Journal of Computing*, 33(3), 544-562, 2003.