Local Search Algorithms for k-Median*[1]

- In a previous lecture, we saw a 3-approximate local search for metric facility location. This note shows how a very similar local search gives a 5-approximation for the k-median problem. We have tried to keep this note self-contained, although we may still refer to the previous lecture from time to time.

  In the k-median problem we are given a set F of facilities, a set C of clients, and a metric $d(\cdot,\cdot)$ in $F \cup C$. The objective is to open at most k facilities, that is $X \subseteq F$ with $|X| = k$, and connect clients via assignment $\sigma : C \to X$ to nearest open facility, to minimize

$$\texttt{cost}(X) = \sum_{j \in C} d(\sigma(j), j) \tag{1}$$

- **Local Search for k-median.** The algorithm is the obvious one; we open an arbitrary collection of k facilities, and try to find swaps which decreases cost, stopping when no such swap is possible.

```
1: procedure kMED-LOCAL SEARCH(F, C, d):
2:     X be an arbitrary subset of k facilities.
3:     ▷ Throughout cost(X) is defined using (1) where f_i = 0
4:     while true do:
5:         (Swap): If there exists i ∈ X and i' ∈ F\X such that cost(X−i+i') < cost(X);
       X ← X − i + i'.
6:         Otherwise, break
```

- ***Analysis.*** We prove the following theorem.

Theorem 1. kMED-LOCAL SEARCH is a 5-approximation algorithm.

- We use notation similar to that in the case of UFL. Let X be the set of facilities opened at the end of the above algorithm. Let $\sigma(j)$ denote the facility in X client j is connected to. Let $\Gamma(i)$ denote the set of clients connected to facility $i \in X$. Let X^* denote the set of facilities opened in the optimal solution. Let σ^* and Γ^* be defined similarly. Let $d_j := d(\sigma(j), j)$ and $d^*_j := d(\sigma^*(j), j)$ be the connection costs for client j in the algorithm and optimum solution, respectively. Thus, $C_{\mathsf{alg}} = \sum_{j \in C} d_j$ and $C^* = \sum_{j \in C} d^*_j$.

- As in the case of UFL, we need the concepts of `nearest` and its "inverse".

  Fix an $i \in X$. When we close i, we need to figure out how to reassign $\Gamma(i)$. It would be great if $j \in \Gamma(i)$ can be assigned to $i^* := \sigma^*(j)$, but that facility may not be opened. So one tries the next best thing : open the nearest facility to this i^*. This motivates the following key definition.

[1] *Lecture notes by Deeparnab Chakrabarty. Last modified : 14th January, 2022*
These have not gone through scrutiny and may contain errors. If you find any, or have any other comments, please email me at deeparnab@dartmouth.edu. Highly appreciated!

Given $i^* \in X^*$, let $\texttt{nearest}(i^*)$ denote the facility i in X with minimum $d(i, i^*)$.

For any facility $i \in X$, define

$$X_i^* := \{i^* \in X^* : \texttt{nearest}(i^*) = i\}. \tag{2}$$

that is, the facilities in X^* for which i is the closest facility. In some sense, it is the "inverse" of the $\texttt{nearest}$ map, and indeed would exactly be that if $\texttt{nearest}$ was a bijection. Instead, X_i^* maps to a subset of facilities in X^*. Crucially note that by definition, $X_i^* \cap X_{i'}^*$ for any two facilities in X. See Figure 1 for an illustration

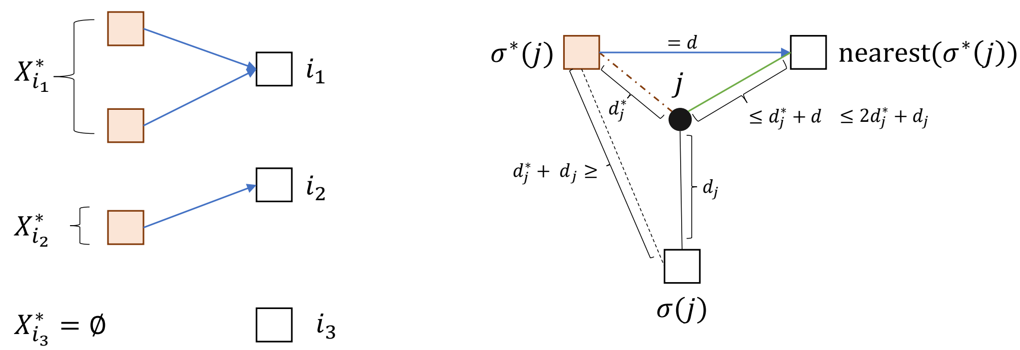

Figure 1: *Salmon squares denote facilities in X^* while empty squares denote facilities in X. The blue arrows denote the* $\texttt{nearest}$ *map from X^* to X. The sets X_i^* for each $i \in X$ is denoted; note that $X_{i_1}^*$ has two facilities, $X_{i_2}^*$ has 1, while $X_{i_3}^*$ is empty. The right figure illustrates Claim 1.*

Here is a useful fact which follows easily from triangle inequality and definition of $\texttt{nearest}$ (see Figure 1 for an illustration).

Claim 1. For any $j \in C$, $d(\texttt{nearest}(\sigma^*(j)), j) \leq d_j + 2d_j^*$.

Proof. Let j be assigned to i in σ and i^* in σ^*. Then, triangle inequality implies $d(\texttt{nearest}(i^*), j) \leq d(i^*, j) + d(\texttt{nearest}(i^*), i^*) \leq d_j^* + d(i, i^*)$, where the last inequality is by definition of $\texttt{nearest}(i^*)$. Triangle inequality again implies $d(i^*, i) \leq d(i, j) + d(i^*, j)$. □

- ***A Wishful thinking.*** Suppose for the time being that $\texttt{nearest}$ was indeed a bijection. That is, for every $i \in X$, X_i^* is a singleton. Then consider swapping i and the unique facility $i^* \in X_i^*$. Consider the following reassignment : all the clients $j \in \Gamma^*(i^*)$ are re-assigned to i^*, and all the clients $j \in \Gamma(i)$ are reassigned to $\texttt{nearest}(\sigma^*(j))$. Note that this is possible since either $\sigma^*(j) \neq i^*$ in which case its $\texttt{nearest}(\sigma^*(j))$ is in $X \setminus i$, or $\sigma^*(j) = i^*$ and it has been already re-assigned to i^* when we reassigned $\Gamma^*(i^*)$. See Figure 2 for an illustration. By Claim Claim 1, the increase in cost due to reassignment of $j \in \Gamma(i) \setminus \Gamma^*(i^*)$ is at most $2d_j^*$. Thus, the difference due to this reassignment is

$$\sum_{j \in \Gamma^*(i^*)} (d_j^* - d_j) + \sum_{j \in \Gamma(i) \setminus \Gamma^*(i^*)} 2d_j^* \underbrace{\geq}_{\text{local optimality}} 0 \tag{3}$$

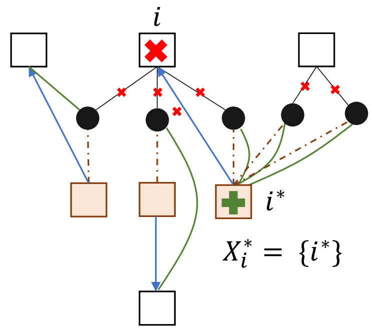

Figure 2: *Salmon squares denote facilities in X^* while empty squares denote facilities in X. Dotted brown lines denote the assignment σ^*. The blue arrows denote the* `nearest` *map from X^* to X. Green lines denote reassignments. In the figure, $X_i^* = \{i^*\}$ and we swap i and i^*. All $j \in \Gamma^*(i^*)$ are reassigned to i^*. For all $j \in \Gamma(i) \setminus \Gamma^*(i^*)$, we must have* $\texttt{nearest}(\sigma^*(j)) \in X \setminus i$, *and they are reassigned to that facility.*

If we now add this over all (i, i^*) pairs with $i \in X$ and $X_i^* = \{i^*\}$, then we would get

$$\sum_{i^* \in X^*} \sum_{j \in \Gamma^*(i^*)} (d_j^* - d_j) + \sum_{i \in X} \sum_{j \in \Gamma(i) \setminus \Gamma^*(i^*)} 2d_j^* \underbrace{\geq}_{\text{local optimality}} 0$$

In the first summation in the LHS above, every client $j \in C$ participates exactly once. In the second summation, every client $j \in C$ participates at most once. Therefore,

$$\sum_{j \in C} (d_j^* - d_j) + \sum_{j \in C} 2d_j^* \geq 0 \quad \Rightarrow \quad 3\mathsf{opt} := 3\sum_{j \in C} d_j^* \geq \sum_{j \in C} d_j =: \mathsf{alg}$$

and we would have a 3-approximation.

- However, the `nearest` map may not be a bijection. And therefore, we need to work a bit more (at the cost of the approximation factor).

  Let $X_0 := \{i \in X : |X_i^*| = 0\}$, $X_1 := \{i \in X : |X_i^*| = 1\}$, and $X_2 := \{i \in X : |X_i^*| \geq 2\}$. In Figure 1 left side, we have $X_0 = \{i_3\}$, $X_1 = \{i_2\}$, and $X_2 = \{i_1\}$. The above bullet point shows if $X_1 = X$, then we would get a 3-approximation. It is instructive, however, to try and see where the above argument fails. That is, if we write (3) for (i, i^*) for all $i \in X_1$ and then try to sum up, where do we fall short? One sees that we don't account the d_j's for all clients, rather only for the clients in the $\Gamma^*(i^*)$'s seen. In particular, if a facility $i' \in X^*$ is *not* in X_i^* for any $i \in X_1$, then we have not been able to argue about the clients in $\Gamma^*(i')$. The next idea defines "swap pairs" such that every facility of X^* is involved in such a pair.

- ***Swap Pairs.*** We describe a set $R \subseteq X^* \times X$ with $|R| = k$ which will be the potential swaps we consider. We need them to have the following properties.

  a. For all $i^* \in X^*$, there exists *exactly one* $i \in X_0 \cup X_1$ such that $(i^*, i) \in R$.
  b. For every $i \in X_1$ there is *exactly one* $i^* \in X^*$ with $(i^*, i) \in R$.

c. For every $i \in X_0$ there is *at most two* $i^* \in X^*$ with $(i^*, i) \in R$.

In other words, we can think of R as a bipartite graph from X^* to $X_0 \cup X_1$, then the degree $\deg(i)$ of every vertex i in X^* and X_1 is 1 and the degree $\deg(i)$ of every vertex in X_0 is ≤ 2.

Indeed, this is easy. For all $i \in X_1$, let i^* be the unique element in X_i^*. We add (i^*, i) to R. Now the remaining $k - |X_1|$ facilities of X^* need to be mapped to X_0. Since $k - |X_1| = |X_0| + |X_2| \leq 2|X_0|$, we can always find one such that every $i \in X_0$ is mapped with at most 2 facilities in X^*. An arbitrary one will do. See Figure 3 for an illustration.

- ***The full proof.*** Consider now the swaps defined by R: for $(i^*, i) \in R$, add i^* in and delete i. For each $j \in \Gamma^*(i^*)$, we re-assign it to i^*. By design, for every $j \in \Gamma(i) \setminus \Gamma^*(i^*)$, we have $\texttt{nearest}(\sigma^*(j)) \in X - i + i^*$. Note that, by Claim 1, these j's would pay at most $d_j + 2d_j^*$. Since swaps don't decrease cost, we get that for all $(i^*, i) \in R$, (3) holds. That is,

$$\sum_{j \in \Gamma^*(i^*)} (d_j^* - d_j) + \sum_{j \in \Gamma(i) \setminus \Gamma^*(i^*)} 2d_j^* \underbrace{\geq}_{\text{local optimality}} 0$$

  Summing over all pairs in R, we get

$$\sum_{(i^*,i) \in R} \sum_{j \in \Gamma^*(i^*)} (d_j^* - d_j) + \sum_{(i^*,i) \in R} \sum_{j \in \Gamma(i) \setminus \Gamma^*(i^*)} 2d_j^* \geq 0$$

  The first summation is precisely $\sum_{i^* \in X^*} \deg(i^*) \cdot \left(\sum_{j \in \Gamma^*(i^*)} (d_j^* - d_j)\right) = C^* - C_{\mathsf{alg}}$ since $\deg(i^*) = 1$ for all $i^* \in X^*$ and each $j \in C$ appears in exactly one $\Gamma^*(i^*)$. The second summation is precisely $2 \sum_{i \in X_0 \cup X_1} \deg(i) \cdot \left(\sum_{j \in \Gamma(i)} d_j^*\right)$ which is at most $4C^*$ since $\deg(i) \leq 2$ and each $j \in C$ appears in at most one $\Gamma(i) \setminus \Gamma^*(i^*)$. Therefore, the LHS is at most $5C^* - C_{\mathsf{alg}}$, and thus we get that $5C^* \geq C_{\mathsf{alg}}$. This completes the proof of Theorem 1.

Notes

The local search algorithm described above is from the paper [1] by Arya, Garg, Khandekar, Meyerson, Munagala, and Pandit. The analysis here is inspired by the simpler analysis in [4] by Gupta and Tangwongsan. For k-median, one can look at p-swaps where p-facilities are swapped out; we have investigated the $p = 1$ case. It is not too hard to generalize the above analysis to prove that it gives a $\left(3 + \frac{2}{p}\right)$-approximation. The runtime becomes $n^{O(p)}$. The analysis is tight and an example can be found in [1]. This factor, was the best known factor for k-median for close to a decade, till the paper [5] by Li and Svensson gave a $(1 + \sqrt{3}) \approx 2.732$-approximation using different methods. The best known approximation factor of 2.675 is in the paper [2]. Very recently, a *non-oblivious* local search method was announced in the paper [3] and was analyzed to have a factor ≤ 2.836. This is not known to be tight.

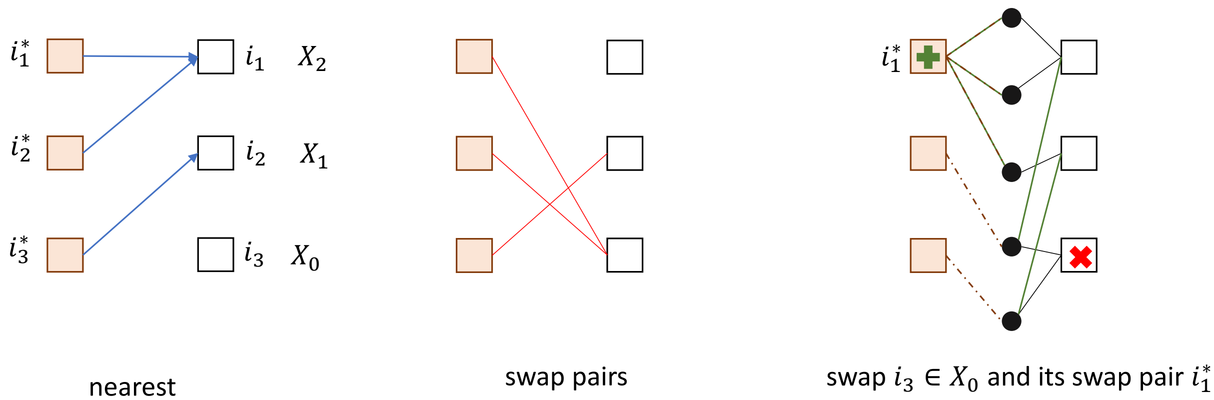

Figure 3: *The first figure shows the* `nearest` *relation. The middle red lines show swap pairs. The third shows a swap of a facility in X_0 with its swap pair. Salmon squares denote facilities in X^* while empty squares denote facilities in X. Dotted brown lines denote the assignment σ^*. The blue arrows denote the* `nearest` *map from X^* to X. Green lines denote reassignments.*

References

[1] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit. Local search heuristics for k-median and facility location problems. *SIAM Journal on Computing (SICOMP)*, 33(3):544–562, 2004.

[2] J. Byrka, T. Pensyl, B. Rybicki, A. Srinivasan, and K. Trinh. An improved approximation for k-median, and positive correlation in budgeted optimization. In *Proc., ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 737–756, 2014.

[3] V. Cohen-Addad, A. Gupta, L. Hu, H. Oh, and D. Saulpic. An improved local search algorithm for k-median. *arXiv preprint arXiv:2111.04589*, 2021. To appear in SODA 2022.

[4] A. Gupta and K. Tangwongsan. Simpler analyses of local search algorithms for facility location. *arXiv preprint arXiv:0809.2554*, 2008.

[5] S. Li and O. Svensson. Approximating k-median via pseudo-approximation. *SIAM Journal on Computing (SICOMP)*, 45(2):530–547, 2016.