CS787: Advanced Algorithms

Topic: Local Search Algorithms **Presenter(s):** Theodora Hinkle, Ning Zhang

In this note, we will survey the **local search heuristics** for the metric **k-median** and **facility location problems**. We define the *locality gap* of a local search heuristics for a minimization problem as the maximum ratio of *local* optimal solution (produced by local search heuristics) to the *global* optimal solution. For k-median, the local search with single swap has a locality gap of 5. Moreover, if we allow p facilities to be swapped simultaneously, the locality gap will be $3+\frac{2}{p}$. For (uncapacitated) facility location, the local search, which permits adding, dropping, and swapping a facility, has a locality gap of 3. All above results are currently the best ones by using local search heuristics.

17.2.1 Introduction

17.2.1.1 Background for k-median and facility location problems

Facility location problems capture a common need of many real world businesses: to decide where to locate their facilities in a way that most effectively serves their clients. Many aspects of the real world problem make finding solutions difficult, and even the simplified models of metric k-median and metric uncapacitated facility location are hard problems.

A facility location problem generally has a set of possible facility locations and a set of clients to be served. There can be distances defined between the clients and facilities, leading to a measure of effectiveness of a solution can vary depending on the **service cost**, the distance between clients and the facilities they are assigned to in the solution. There can also be unequal costs to opening different facilities, leading to a measure of effectiveness depending on the **facility cost**, the total cost to open the facilities chosen by a solution.

Using various combinations of these two measures as objective functions to be minimized leads to interesting problems. In metric k-median, the number of facilities which may be open is at most k and the total service cost to all clients is minimized. In metric uncapacitated facility location, the sum of the total facility cost and service cost is minimized, and there are no restrictions on the number of clients any facility can serve. We will formally define these two problem shortly.

17.2.1.2 Background for local search heuristics

Local search is a metaheuristic for solving computationally hard optimization problems. Local search can be used on problems that can be formulated as finding a solution maximizing a criterion among a number of candidate (feasible) solutions. Local search algorithms move from solution to solution in the space of candidate solutions (the search space) until the local optimal solution is found (or a time bound is elapsed). Now, the question is how local search algorithm starts from a candidate solution and then iteratively moves to a better solution? This is only possible if a **neighborhood relation** is defined on the search space. As an example, the neighborhood of a vertex cover could be defined as another vertex cover only differing by one node.

Now, we define the local search algorithm (Algorithm 1) formally. A generic local search algorithm can be described by a set S^* of all feasible solutions, a cost function c: $S^* \to \mathbb{R}$, a *neighborhood* structure N: $S^* \to 2^{S^*}$, and an oracle that, given any solution S, finds a solution $S' \in N(S)$ such that $c(S') < c(S)$. A solution $S \in S^*$ is called **local optimal** if $c(S) \leq c(S')$ for all $S' \in N(S)$. For example, Algorithm 1 always returns the local optimal solution. The cost function and neighborhood structure N will be different for different problems and algorithms.

Algorithm 1 Local Search Algorithm

S is an arbitrary feasible solution in S^*
while $\exists S' \in N(S)$ such that $c(S') < c(S)$ **do**
  $S \leftarrow S'$
end while
return S

For an instance I of a minimization problem, let $global(I)$ denote the cost of the global optimum and $local(I)$ be the cost of a locally optimum solution provided by a certain local search heuristic. We call the supremum of the ratio $local(I)/global(I)$ the **locality gap** of this local search procedure. Our proof of the locality gap proceeds by considering a suitable, polynomially large subset $Q \subseteq N(S)$ of neighboring solutions and arguing that

$$\sum_{S' \in Q} (c(S') - c(S)) \leq \alpha \cdot c(O) - c(S)$$

where O is the global optimal solution and $\alpha > 1$ is a constant. This implies that $c(S) \leq \alpha \cdot c(O)$, which α gives the locality gap.

17.2.2 Notations and Problem formulation

In the k-median and facility location problem, we are given two sets: F, the set of *facilities*, and C, the set of clients. Let $c_{ij} \geq 0$ denote the cost of serving client $i \in C$ by a facility $j \in F$; we will think of this as the distance between client i and facility j. The goal in these problems is to identify a subset of facilities $S \subseteq F$ and to serve all clients by facilities in S such that some objective function is minimized. The facilities in S are said to be open. The metric versions of these problems assume that distances c_{ij} are symmetric and satisfy the triangle inequality. The problems considered in this paper are defined as follows.

1. **metric k-median problem**: Given integer k, identify a set $S \subseteq F$ of at most k facilities to open such that the total cost of serving all clients by open facilities is minimized.

2. **metric uncapacitated facility location (UFL) problem**: For reach facility $i \in F$, given a cost $f_i \geq 0$ of opening facility i. The goal is to identify a set of facilities $S \subseteq F$ such that the total cost of opening the facilities in S and serving all the clients by open facilities is minimized.

17.2.3 k-median problem and analysis

17.2.3.1 Local search with single swap

In this section, we consider a local search using single swap. A swap is effected by closing a facility $s \in S$ and opening a facility $s' \in F$ and is denoted by $< s, s' >$; hence the neighborhood of S is defined as $B(S) = \{S - s + s' | s \in S\}$. We start with an arbitrary set of k facilities S and keep improving the solution S with a single swap at a time until we reach a local optimal solution (we can not improve the solution by a single swap). The algorithm is described in Algorithm 1.

17.2.3.2 Analysis

Let S be the local optimal solution returned by the local search and O be a global optimal solution. We will show that this local search has a locality gap of 5, that is $c(S) \leq 5 \cdot c(O)$. First, we use $N_S(s)$ to denote the set of clients that facility s serves in S and $N_O(o)$ to denote the set of clients that facility o serves in O. The mathematical presentations are as follows:

$$N_S(s) = \{i \in C, \sigma(i) = s\}$$
$$N_O(o) = \{i \in C, \sigma^*(i) = s\}$$

Here $\sigma^*(i)$ and $\sigma(i)$ are the global and local optimal location of the facility serving the client i respectively. Now we define the "improvement" of a swap when adding $o \in O$ to local solution S, as follows:

$$c(S) - c(S + o) \geq \sum_{i \in N_O(o)} c_{i\sigma(i)} - c_{i\sigma^*(i)} \tag{17.2.3.1}$$

If we consider adding each facility of O individually to S, we have

$$\sum_{o \in O} [c(S) - c(S + o)] \geq c(S) - c(O) \tag{17.2.3.2}$$

Then, we need to know how to bound the impact of dropping a facility of S. Toward this end, we define the notion of **capture**:

Definition 17.2.3.1 *For a local solution S and optimal solution O, we say $s \in S$* ***captures*** *$o \in O$ if at least half of o's clients are served by s, i.e.,*

$$|N_S(s)\bigcap N_O(o)| > \frac{|N_O(o)|}{2}$$

It is easy to see that a facility $o \in O$ is captured by at most one facility in S. We will consider the following two cases when bounding the impact of dropping s.

1. A facility $s \in S$ captures exactly one $o \in O$. In this situation, we just swap this pair $< s, o >$.
2. Some facilities in S capture more than one facilities in O, and some facilities in O may not have been captured by any facility in S. Here none of the facilities $s \in S$ which captures more than one optimal choices should be swapped, instead we will swap the facilities in O with the facilities in S which capture no facilities in O.

We will first show that the local algorithm described above is a 3-approximation algorithm, if all facilities in O were of CASE 1.

The challenge in our analysis of the swapping process is that after swapping, some of the clients served by some facilities in S must be reassigned to other facilities since the original ones have been swapped out. We should find a way to estimate the change of the cost in this reassignment process. So for $i \in N_S(s)\backslash N_O(o)$ in a swapping on $< s, o >$, we use a 1-1 and onto mapping function $\pi : N_O(o) \rightarrow N_O(o)$ satisfying the following property,

Property 17.2.3.2 *Let the node set $N_s^o = N_S(s)\bigcap N_O(o)$. If s does not capture o, then $\pi(N_s^o)\bigcap N_s^o = \emptyset$*

We outline how to obtain one such mapping π. Let $m = |N_O(o)|$. Order the clients in $N_O(o)$ as $c_0, \ . \ . \ . \ , c_{m-1}$ such that for every $s \in S$ with a nonempty N_s^o , the clients in N_s^o are consecutive; that is, there exists p, q, $0 \leq p \leq q \leq m-1$, such that $N_s^o = \{c_p, ..., c_q\}$. Now for client $i \in N_O(o)$, set $\pi(i) = i + \lfloor \frac{m}{2} \rfloor$. The proof of this property is omitted due to the page limits. For the complete proof, please refer to [1].

For a given swapping between $< s, o >$, since S is locally optimal, we have

$$c(S) - c(S + o - s) \leq 0 \tag{17.2.3.3}$$

Consider a swap $< s, o >$. We place an upper bound on the increase in the cost due to this swap by reassigning the clients to the facilities in $S - s + o$ as follows. The clients $j \in N_O(o)$ are now assigned to o. Consider a client i who is served only by s but not o, $i \in N_s^{o'}$ for $o' \neq o$. As s only captures o and does not capture o', by Property 17.2.3.2 of π, we have that $\pi(i) \notin N_S(s)$ and $\pi(i) \in N_O(o')$. Let $j = \pi(i) \in N_S(s')$. Note that the distance that the client i travels to the nearest facility in $S - s + o$ is at most $c_{is'}$. From the triangle inequality, $c_{is'} \leq c_{i\sigma^*(i)} + c_{j\sigma(j)} + c_{j\sigma^*(j)}$. The clients which do not belong to $N_S(s)\bigcup N_O(o)$ continue to be served by the same facility.

We would like to give an intuitive explanation of what we just did. Consider a client i served by facility o in the globally optimum solution O. Suppose in the locally optimum solution S, i is served by facility s. If there is a swap $< s, o >$, i must be assigned to a facility besides s. Thus, the cost of serving i may change. What we did above is to create a mapping π from clients served by o in O, to different clients served by o in O, so that if there is a client like i who loses their original facility, we can say that the new cost of serving them is no greater than routing them to o, then routing them to $\pi(i)$, then routing them to the facility serving $\pi(i)$ in the S after the swap.

Since we need to add o while dropping s, for the clients who are served only by s but not o, the cost will increase since they should be reassigned. So

$$c(S)-c(S+o-s) = \sum_{i\in N_O(o)} (c_{i\sigma(i)}-c_{i\sigma^*(i)})+\sum_{i\in N_S(s), i\notin N_O(o)} (c_{i\sigma(i)}-c_{i\sigma^*(i)}-c_{\pi(i)\sigma(\pi(i))}-c_{\pi(i)\sigma^*(\pi(i))}) \leq 0 \tag{17.2.3.4}$$

Then for the clients $i \in N_S(s) - N_O(o)$, after the swapping of $< s, o >$, i should be reassigned. Here we will map i to $j = \pi(i) \in N_O(o)$, and the cost of reassigning can be estimated,

Due to the triangle inequality, we have,

$$c_{j\sigma(j)} \leq c_{j\sigma^*(j)} + c_{i\sigma^*(i)} + c_{i\sigma(i)}$$

Therefore, we have the following equation,

$$\begin{aligned} c_{i\sigma^*(i)} + c_{j\sigma(j)} + c_{j\sigma^*(j)} - c_{i\sigma(i)} &\leq c_{i\sigma^*(i)} + c_{j\sigma^*(j)} + c_{i\sigma^*(i)} + c_{j\sigma^*(j)} + c_{i\sigma(i)} - c_{i\sigma(i)} \\ &= 2(c_{i\sigma^*(i)} + c_{j\sigma^*(j)}) \end{aligned} \tag{17.2.3.5}$$

From Equations 4 and 5, we can get,

$$\sum_{i\in N_O(o)} (c_{i\sigma(i)} - c_{i\sigma^*(i)}) \leq \sum_{i\in N_S(s), i\notin N_O(o)} 2(c_{i\sigma^*(i)} + c_{j\sigma^*(j)}) \tag{17.2.3.6}$$

For the situation that all facilities in O were of CASE 1, according to the swap policy mentioned above, the facility $s \in S$ and also $o \in O$ will be swapped exactly once, since for each $o \in O$, there is exactly one $s \in S$ which captures it. Thus, after finishing all the swaps, there will totally be at most half of the clients i which should be considered when $i \in N_S(s), i \notin N_O(o)$. And also, $j = \pi(i)$ is a one-to-one mapping, so for the right hand side of Equation 6, we have

$$\sum_{i\in N_S(s), i\notin N_O(o)} 2(c_{i\sigma^*(i)} + c_{j\sigma^*(j)}) \leq 2 \cdot c(O) \tag{17.2.3.7}$$

From Equation 6 and 7, we will get the following by summing over all $o \in O$:

$$c(s) - c(O) \leq 2 \cdot c(O)$$

$$c(s) \leq 3 \cdot c(O)$$

For CASE 2, if the number of facilities in O in CASE 2 are l, there are at least $l/2$ facilities in S which will not capture any of the facilities in S, because otherwise, some facilities in O are captured more than once, which is impossible. We will consider swapping these l facilities in O with the facilities in S which capture no facilities in O and show that the algorithm will produce a 5-approximation solution in this case.

$$\sum_{i \in N_S(s), i \notin N_O(o)} 2(c_{i\sigma^*(i)} + c_{j\sigma^*(j)}) \leq 4 \cdot c(O) \tag{17.2.3.8}$$

Then we combine Equation 6 with 8, we will get

$$4 \cdot c(O) \geq c(S) - C(O)$$
$$c(S) \leq 5 \cdot c(O)$$

The above inequality completes the whole analysis of the local search algorithm with single swap.

17.2.3.3 Local search with multiswaps

In this section, we generalize the algorithm in the above section to consider multiswaps in which up to $p > 1$ facilities can be swapped simultaneously. The neighborhood structure is now defined by

$$B(S) = \{(S \backslash A) \cup B | A \subseteq S, B \subseteq F, |A| = |B| \leq p\}$$

The neighborhood captures the set of solutions obtainable by deleting a set of at most p facilities A and adding a set of facilities B where $|B| = |A|$; this swap will be denoted by $< A, B >$. We prove that the locality gap of the k-median problem with respect to this operation is exactly $(3 + \frac{2}{p})$.

17.2.3.4 Analysis

We extend the notion of capture as follows. For a subset $A \subseteq S$, we define

$$capture(A) = \{o \in O : |N_S(A) \cap N_O(o)| > |N_O(o)/2|\}$$

It is easy to observe that if $X, Y \subseteq S$ are disjoint, then $capture(X)$ and $capture(Y)$ are disjoint and if $X \subset Y$, then $capture(X) \subseteq capture(Y)$. We now partition S into sets $A_1, ..., A_r$ and O into sets $B_1, ..., B_r$ such that for all i, $1 \leq i \leq r-1$, $|A_i| = |B_i|$ and $B_i = capture(A_i)$. We call a facility in S is *bad* if it captures at least one facility in O, and *good* otherwise. Our partition of S would have the property that every $A_i, 1 \leq i \leq r-1$ would have exactly one bad facility; thus $r-1$ equals the number of bad facilities. The set A_r contains only good facilities and it follows $|A_r| = |B_r|$, since $|S| = |O|$. The procedure to obtain such a partition is given in Algorithm 2.

Algorithm 2 Partition

for $i = 1$ to $r - 1$ **do**
  $A_i \leftarrow b$, where $b \in S$ be any bad facility
  $B_i \leftarrow capture(A_i)$
  while $|A_i| \neq |B_i|$ **do**
    $A_i \leftarrow A_i \cup g$, where $g \in S \backslash A_i$ be any good facility
    $B_i \leftarrow capture(A_i)$
  end while
  $S \leftarrow S \backslash A_i$
  $O \leftarrow O \backslash B_i$
end for
$A_r \leftarrow S$
$B_r \leftarrow O$

Property 17.2.3.3 *Algorithm 2 terminates with partitions of S and O, satisfying the properties listed above.*

The proof for Property 3.3 can be found in [1].

Now we define the swaps as follows. If for some i, we have $|A_i| = |B_i| \leq p$ then we consider the swap $< A_i, B_i >$. From the local optimality of S, we have the following inequality

$$c((S \backslash A_i) \cup B_i) - c(S) \geq 0$$

On the other hand, for some i, we have $|A_i| = |B_i| = q \geq p$; we swap each facility $o \in B_i$ with each of the $q - 1$ good facilities $s \in A_i$. Note that if $i \neq r$, there are exactly $q - 1$ good facilities in A_i, and for $i = r$, we select any $q - 1$ out of q good facilities in A_r. For each such swap $< s, o >$, we have,

$$c(S) - c(S + o - s) \leq 0$$

We add such $q(q-1)$ inequalities and multiply them by a factor $1/(q-1)$. Thus, each good facility in A_i is considered in at most $q/(q-1) \leq (p+1)/p$ swaps.

For each facility $o \in O$, $N_O(o)$ is partitioned as follows.

1. let i, $1 \leq i \leq r$, be such that $|A_i| \leq p$, so that the swap $< A_i, B_i >$ is considered above. We consider the part, $p_{A_i} = N_S(A_i) \cap N_O(o)$.

2. let i, $1 \leq i \leq r$, be such that $|A_i| > p$. We consider the parts $p_s = N_S(s) \cap N_O(o)$ for each $s \in A_i$.

As before, for each facility $o \in O$, we consider a 1-1 and onto mapping $\pi : N_O(o) \rightarrow N_O(o)$ with the following property.

Property 17.2.3.4 *For all parts $p = p_{A_i}$ or p_s, such that $|p| \leq \frac{1}{2}|N_O(o)|$, we have $\pi(p) \cap p = \emptyset$.*

As this condition is imposed only on the parts that have at most half the number of clients in $N_O(o)$, such a mapping π exists. While doing a swap $< A_i, B_i >$ (resp. $< s, o >$), we would be able to reassign clients $j \in N_S(A_i) - N_O(B_i)$ (resp. $N_S(s) - N_O(o)$) to the facility $s' \notin A_i$ (resp. $s \neq s'$) that serves $\pi(j)$ in S.

The swaps defined above satisfy the following properties:

1. Each facility in O is considered for a swapped-in exactly once.
2. Each facility in S is considered for a swapped-out at most $(p+1)/p$ times.
3. If a swap $< A, B >$ is considered, $capture(A) \subseteq B$.

Recall that in the single swap analysis, as each facility in S was considered for swapping-out at most twice, we got a $(1 + 2 \times 2)$ approximation. Here, $(p+1)/p$ replaces 2 and using the same argument gives a $(1 + 2 \times (p+1)/p) = 3 + \frac{2}{p}$ approximation.

17.2.4 Uncapacitated Facility Location and Analysis

In this problem, we can open an unlimited number of facilities, but each facility $i \in F$ has a cost $f_i \geq 0$ of opening it. The UFL problem is to identify a subset $S \in F$ and to serve the clients in C by the facilities in S such that the sum of facility (opening) costs and service costs is minimized. That is, if a client $j \in C$ is assigned to a facility $\sigma(j) \in S$, then we want to minimize $c(S) = \sum_{i \in S} f_i + \sum_{j \in C} c_{j\sigma(j)}$. Note that for a fixed S, serving each client by the nearest facility in S minimizes the service cost.

17.2.4.1 Local search algorithm

We present a local search procedure for the metric UFL problem with a locality gap of 3. The operations allowed in a local search step are adding a facility, deleting a facility, and swapping facilities. Hence given a solution S, the neighborhood $N(S)$ is defined by

$$B(S) = \{S + s' | s' \in F\} \cup \{S - s | s \in S\} \cup \{S - s + s' | s \in S, s' \in F\}$$

17.2.4.2 Analysis

For any set of facilities $S \subseteq F$, let $c_f(S) = \sum_{i \in S} f_i$ denote the opening cost of the solution S. Also, let $c_s(S)$ be the total cost of serving the clients in C by the nearest facilities in S. The following bound on the service cost of S has been proved in [2].

Lemma 17.2.4.1 $c_s(S) \leq c_f(O) + c_s(O)$

Proof: Consider an operation in which a facility $o \in O$ is added. Assign all the clients $N_O(o)$ to o. From the local optimality of S we get $f_o + \sum_{j \in N_O(o)} (c_{j\sigma^*(j)} - c_{j\sigma(j)}) \geq 0$. If we add such inequalities for every $o \in O$, we get the result in the lemma.

□

The following lemma gives a bound on the facility (opening) cost of S

Lemma 17.2.4.2 $c_f(S) \leq c_f(O) + 2 \cdot c_s(O)$

Proof: As before, we assume that for a fixed $o \in O$, the mapping $\pi : N_O(o) \to N_O(o)$ is a 1-1 and onto mapping and satisfies Property 3.2. In addition, we assume that if $|N_s^o| > \frac{1}{2}|N_O(o)|$, then for all $j \in N_s^o$ for which $\pi(j) \in N_s(s)$, we have $\pi(\pi(j)) = j$ and such a mapping is easy to find. Recall that a facility $s \in S$ is called *good* if s does not capture any o, that is, $\forall o \in O$, $|N_s^o| \leq \frac{1}{2}|N_O(o)|$.

The opening cost of good facilities can be bounded easily as follows. Consider an operationin which a good facility $s \in S$ is dropped. Let $j \in N_s(s)$ and $\pi(j) \in N_s(s')$. Since s is good - does not capture any facility $o \in O$ - we have $s \neq s'$. If we assign j to s', we have $-f_s + \sum_{j \in N_S(s)} (c_{j\sigma^*(j)} + c_{\pi(j)\sigma^*(\pi(j))} + c_{\pi(j)\sigma(\pi(j))} - c_{j\sigma(j)}) \geq 0$, which can be simplified as $-f_s + \sum_{j \in N_S(s)} (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j) \geq 0$. Since for all $j \in N_S(s)$, $\pi(j) \neq j$, then $\sum_{j \in N_S(s), \pi(j)=j} O_j$ is zero and hence we can rewrite the above inequality as,

$$-f_s + \sum_{j \in N_S(s)} (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j + 2 \sum_{j \in N_S(s), \pi(j)=j} O_j \geq 0 \qquad (17.2.4.9)$$

For bounding the opening cost of a bad facility $s \in S$ we proceed as follows. Suppose s captures the facilities $P \in O$. Let $o \in P$ be the facility nearest to s. We consider the swap $< s, o >$. The clients $j \in N_S(s)$ are now reassigned to the facilities in $S - s + o$ as follows.

1. Suppose $\pi(j) \in N_s(s')$ for $s \neq s'$. Then j is assigned to s'. Let $j \in N_O(o')$, we have $c_{js'} \leq c_{jo'} + c_{\pi(j)o'} + c_{\pi(j)s'} = O_j + O_{\pi(j)} + S_{\pi(j)}$
2. Suppose $\pi(j) = j \in N_S(s)$ and $j \in N_O(o)$. Then j is assigned to o.
3. Suppose $\pi(j) = j \in N_S(s)$ and $j \in N_O(o')$ for $o' \neq o$. By Property 3.2 of mapping π, s captures o' and hence $o' \in P$. Then client j is now assigned to o. From the triangle inequality, $c_{jo} \leq c_{js} + c_{so}$ Since $o \in P$ is the closest facility to s, so we have $c_{so} \leq c_{so'} \leq c_{js} + c_{jo'}$. Therefore, $c_{jo} \leq c_{js} + c_{js} + c_{jo'} = S_j + S_j + O_j$

Thus for swap $< s, o >$ we get the following inequality

$$f_o - f_s + \sum_{j \in N_S(s), \pi(j) \neq j} (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j) + \sum_{j \in N_O(o), \pi(j) \in N_S(s)} (O_j - S_j) + \sum_{j \notin N_O(o), \pi(j) \in N_S(s)} (S_j + S_j + O_j - S_j) \geq 0 \qquad (17.2.4.10)$$

Now consider an operation in which a facility $o' \in P - o$ is added. The clients $j \in N_O(o')$ for which $\pi(j) = j \in N_s(s)$ are now assigned to o', and this yields the following inequality.

$$f_{o'} + \sum_{j \in N_O(o'), \pi(j)=j \in N_S(s)} (O_j - S_j) \geq 0 \qquad (17.2.4.11)$$

Adding Equation 10 to Equation 11 we get, for a bad facility $s \in S$,

$$\sum_{o' \in P} f_{o'} - f_s + \sum_{j \in N_S(s), \pi(j) \neq j} (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j) + 2 \sum_{j \in N_S(s), \pi(j)=j} O_j \geq 0 \qquad (17.2.4.12)$$

The last term on the left is an upper bound on the sum of the last two terms on the left of Equation 10 and the last term on the left of the Equation 11 added for all $o' \in P - o$.

Now we add Equation 5 for all good facilities $s \in S$, Equation 12 for all bad facilities $s \in S$, and inequalities $f_o \geq 0$ for all $o \in O$, which are not captured by any $s \in S$, to obtain

$$\sum_{o \in O} f_o - \sum_{s \in S} f_s + \sum_{\pi(j) \neq j} (O_j + O_{\pi(j)} + S_{\pi(j)} - S_j) + 2 \sum_{\pi(j)=j} O_j \geq 0$$

Note that since $\sum_{j:\pi(j) \neq j} O_j = \sum_{j:\pi(j) \neq j} O_{\pi(j)}$ and $\sum_{j:\pi(j) \neq j} S_j = \sum_{j:\pi(j) \neq j} S_{\pi(j)}$, we have $\sum_{j:\pi(j) \neq j}(O_j + O_{\pi(j)} + S_{\pi(j)} - S_j) = 2\sum_{j:\pi(j) \neq j} O_j$ and hence $c_f(O) - c_f(S) + 2 \cdot c_s(O) \geq 0$. This proves the desired lemma. □

Combining Lemma 4.1 and 4.2, we have the following result.

Theorem 17.2.4.3 *The local search algorithm for the metric UFL problem with the neighborhood structure $N(S) = \{S + s'\} \cup \{S - s | s \in S\} \cup \{S - s + s' | s \in S\}$ has a locality gap of at most 3.*

References

[1] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala and V. Pandit, Local search heuristic for k-median and facility location problems, In *SIAM J. Comput.*, Vol. 33, No. 3, pp, 544-562, 2004.

[2] M. Korupolu, C. Plaxton, and R. Rajaraman, Analysis of a local search heuristic for facility location problems, In *SODA*, pp, 1-10, 1998.