RAIRO-Theor. Inf. Appl. 49 (2015) 285–306
DOI: 10.1051/ita/2016002

Available online at:
www.rairo-ita.org

ANALYSIS OF A LOCAL SEARCH ALGORITHM FOR THE k-FACILITY LOCATION PROBLEM

Nasim Samei[1] and Roberto Solis-Oba[2,*]

Abstract. In the k-facility location problem we are given the possible locations for a group of at most k facilities that are to provide some service to a set of clients. Each location has an associated cost for building a facility there. The goal is to select the locations for the facilities that minimize the sum of the cost for building the facilities and the total cost for servicing the clients. In this paper we analyse a local-search heuristic with multiple swaps for the metric k-facility location problem and prove that it has locality gap of $2+\sqrt{3}+\epsilon$ for any constant $\epsilon > 0$. This matches the bound obtained by Zhang [*Theoret. Comput. Sci.* **384** (2007) 126–135.] for a local search algorithm that uses insertions and deletions in addition to swaps. We also prove a second, tight, bound for the locality gap of our algorithm which is better than the above one in many cases. For example, when the ratio between the highest and lowest facility cost is bounded by $p+1$, where p is the maximum number of facilities that can be exchanged in a swap operation, the locality of our algorithm is $3+\frac{2}{p}$; this matches the locality gap of the algorithm of Arya *et al.* [*SIAM J. Comput.* **33** (2004) 544–562.] for the k-median problem.

Mathematics Subject Classification. 68W25, 68W40.

Keywords and phrases. Approximation algorithm, local search, k-facility location.

[1] Department of Computer Science, Western University, London, Ontario, Canada

[2] Department of Computer Science, Western University, London, Ontario, Canada.
solis@csd.uwo.ca

* *Research partially supported by the Natural Sciences and Engineering Research Council of Canada, grant 04667-2015.*

1. Introduction

In the k-facility location problem we are given set F of facilities, a set C of clients and an integer value $k > 0$. Each facility $j \in F$ has an opening cost f_j and if facility j is opened it can serve client $i \in C$ at cost c_{ij}. The goal is to select a subset S of at most k facilities that minimizes the cost of serving all the clients plus the cost of opening the facilities in S. Let $\sigma(j)$ represent the facility in S that serves client j; then the goal is to minimize

$$\text{cost}(S) = \Sigma_{i \in S} f_i + \Sigma_{j \in C} c_{j\sigma(j)}.$$

Let $\text{cost}_s(S) = \Sigma_{j \in C} c_{j\sigma(j)}$ and $\text{cost}_f(S) = \Sigma_{i \in S} f_i$ denote the service cost and facility cost of solution S respectively; then $\text{cost}(S) = \text{cost}_s(S) + \text{cost}_f(S)$.

If the service costs satisfy the triangle inequality, the problem is known as the metric k-facility location problem. If we eliminate the constraint on the number of facilities, the problem is called the facility location problem. Another special case of the k-facility location problem is when all the facility costs are zero, then the problem is known as the k-median problem. The k-facility location problem and its variants have applications in a large number of areas, such as banking [5], distributed systems [9], web services [11] and network design [2].

The facility location, k-median and k-facility location problems are known to be NP-hard. Therefore, extensive research has been done on designing approximation algorithms for these problems. For the metric k-facility location problem Jain and Vazirani [7] obtained a 6-approximation algorithm using a primal-dual technique; this approximation ratio was improved to 4 by Jain *et al.* [8] using a dual fitting technique, and later Zhang [13] used a local search approach to improve the approximation ratio to $2 + \sqrt{3} + \epsilon$ for any constant $\epsilon > 0$. For the metric facility location problem Shmoys *et al.* [12] obtained the first constant approximation algorithm by using a linear programming-based technique. Jain and Vazirani [7] obtained a better result using a primal-dual technique yielding an algorithm with approximation ratio 3. The currently best known algorithm for the problem is by Li [10] with approximation ratio 1.488. For the metric k-median problem Charikar *et al.* [4] used a linear programming-based technique to design the first constant ratio approximation algorithm. Charikar and Guha [3] combined a primal-dual technique with a greedy approach and designed an improved algorithm with approximation ratio 4. Arya *et al.* [1] utilized a local search heuristic to design an algorithm with approximation ratio $3+\epsilon$ for any constant $\epsilon > 0$.

1.1. Contributions

In this paper we focus on the metric k-facility location problem and show that a local search approach in which the only allowed operation is multi-swaps, where we can simultaneously swap $p \geq 1$ facilities in the solution with p facilities not in the solution, has approximation ratio $\max\{3, 5 - 2\frac{p-1}{q-1}\}$, where q is a parameter whose value depends on the instance and it will be defined in Section 3.2. For those instances when q is close to p the approximation ratio is close to 3. We present an example showing the tightness of our bound. Using scaling we get a second bound for the approximation ratio of our local search algorithm. This bound, $2 + \frac{1}{p} + \sqrt{3 + \frac{2}{p} + \frac{1}{p^2}}$, matches the bound of the local search algorithm of Zhang [13], which uses insertions and deletions in addition to swaps.

Our local search model is simpler than the one used by Zhang, as it uses only swaps; therefore, our algorithm always considers solutions of the same size, and as a result the search space that our algorithm explores is smaller than the one defined by Zhang's model. It is interesting that our algorithm with its more restricted set of operations achieves the same performance as that of Zhang's algorithm which makes use of a richer set of operations. As a result of using a simpler model our analysis is also simpler than that in [13]. Furthermore, with some minor changes our algorithm could find approximate solutions with the same above approximation ratio for all instances of the k'-facility location problem with $k' < k$. Our first bound is better than the second one when $q \leq (1 + \frac{\sqrt{3}}{3})p - \frac{\sqrt{3}}{3}$. In addition, for the special case when the ratio of the largest facility cost to the smallest facility cost is less than $p + 1$ our first bound reduces to $3 + \frac{2}{p}$ the same approximation ratio the algorithm of Arya *et al.* [1] for the k-median problem.

1.2. Organization of the paper

The rest of the paper is organized in the following way. In Section 2 we propose a local search algorithm for the k-facility location problem that uses multi-swap operations. In Section 3 we analyse the local optimal solution produced by our algorithm and compute the first upper bound for its approximation ratio. In Section 4 we present an example showing the tightness of the bound. In Section 5 we present a different analysis of the algorithm and show that its approximation ratio matches that of Zhang's algorithm.

2. A local search algorithm with multiple swaps

Let S be a set of at most k facilities. We present below a local search algorithm for the metric k-facility location problem based on the following multi-swap operation:

$$\texttt{swap <A,B> :=}(S \setminus A) \cup B$$

where $A \subseteq S$, $B \subseteq F \setminus S$, and $|A| = |B| \leq p$, for a constant $p \geq 1$.

Algorithm 1. Local-Search (F, C, k).

1: **Input:** Set of facilities, set of clients and integer value k
2: **Output:** Local optimum solution
3: $S \leftarrow$ any set of k facilities from F
4:
5: **for** $i \leftarrow 1$ **to** k **do**
6: $\quad S' \leftarrow$ any subset of i facilities from F
7:
8: $\quad$ **while** $\exists$ a multi-swap operation $\langle A, B\rangle$ for S' such that $\text{cost}((S' \setminus A) \cup B) < \text{cost}(S')$ **do**
9: $\quad\quad S' \leftarrow (S' \setminus A) \cup B$
10: $\quad$ **end while**
11: $\quad$ **if** $\text{cost}(S) > \text{cost}(S')$ **then**
12: $\quad\quad S \leftarrow S'$
13: $\quad$ **end if**
14: **end for**
15: return S

In steps 8–10 algorithm Local-Search repeatedly tries to improve on the current solution S' by performing multi-swap operations. This process continues until no multi-swap operation can further improve the cost of the solution; therefore the algorithm finds a local optimal solution of size i for each $1 \leq i \leq k$. At the end the algorithm returns the local optimal solution S of minimum cost.

Let S^* be an optimal solution, where $|S^*| = l \leq k$. Let S^f be the final set of facilities selected by Local Search and let S_i be the set selected by the algorithm for each $i = 1, \dots, k$. Since $\text{cost}(S^f) \leq \text{cost}(S_l)$, if we could prove that $\text{cost}(S_l) \leq \alpha\,\text{cost}(S^*)$ for some value α then we would have proven that $\text{cost}(S^f) \leq \alpha\,\text{cost}(S^*)$; thus showing that the approximation ratio of algorithm Local-Search is α. We, of course, do not know the value of l and that is why the "for" loop in the algorithm tries all possible values for l. We show that for all integers $i = 1, \dots, k$ there exists a value $\alpha > 0$ for which $\text{cost}(S_i) \leq \alpha\,\text{cost}(S_i^*)$, where S_i^* is an optimal solution that uses i facilities. This will prove that $\text{cost}(S_l) \leq \alpha\,\text{cost}(S_l^*) = \alpha\,\text{cost}(S^*)$, and so $\text{cost}(S^f) \leq \alpha\,\text{cost}(S^*)$. Therefore, without loss of generality, in the sequel we analyse only the case when the local optimal solution and global optimal solution have the same size.

The *locality gap* of a local search algorithm for a minimization problem is defined as the maximum ratio of the value of any local optimum solution produced by the algorithm to the corresponding global optimum value. The locality gap of Local Search is then equal to its approximation ratio.

To compute the locality gap of algorithm Local-Search we will consider a set Q of swap operations involving the facilities in the local optimum solution S and facilities from a global optimum solution S^*. Since S is a local optimum solution, then for each swap operation $\langle A_i, B_i\rangle \in Q$, where $A_i \in S$ and $B_i \in S^*$,

$$\text{cost}((S \setminus A_i) \cup B_i) \geq \text{cost}(S). \tag{2.1}$$

Note that algorithm Local-Search might not run in time that is polynomial in the size of the input as every iteration of the "while" loop might only provide a marginal improvement in the cost of the solution so it might require a very large number of iterations to find a local optimum solution. We can proceed as in [1] to ensure a polynomial running time: Replace the condition of the "while" loop as follows:

while $\exists$ a multi-swap operation $\langle A, B\rangle$ for S' such that $\text{cost}((S' \setminus A) \cup B) \leq (1 - \frac{\epsilon}{|Q|})\text{cost}(S')$ **do**

where $\epsilon > 0$ is a constant. Every iteration of this loop decreases the value of the solution by at least a factor of $\frac{\epsilon}{|Q|}$, hence the total number of iterations is at most

$$\frac{\log(\text{cost}(S)) - \log(\text{cost}(S^*))}{\log(\frac{|Q|}{|Q|-\epsilon})} \leq \frac{\log(k) + \log(f_{\max}) + \log(n) + \log(c_{\max})}{\log(\frac{|Q|}{|Q|-\epsilon})}$$

where $f_{\max} = \max\{f_i \mid i \in F\}$, $n = |C|$ and $c_{\max} = \max\{c_{ji} \mid i \in F, j \in C\}$. The above inequality holds because $\text{cost}(S) \leq k\, f_{\max} + n\, c_{\max}$ and without loss of generality we can assume $\text{cost}(S^*) \geq 1$.

In the following sections we will show that

$$0 \leq \sum_{\langle A_i, B_i\rangle \in Q} [\text{cost}((S \setminus A_i) \cup B_i) - \text{cost}(S)] \leq \alpha\, \text{cost}(S^*) - \text{cost}(S) \tag{2.2}$$

for some constant α. Therefore, the locality gap of algorithm Local-Search is α. The set Q that we consider contains no more than $k^2 + k$ multi-swap operations as explained in Sections 3.2 and 4.2, so the total number of iterations performed by Local-Search is at most

$$[\log(k) + \log(f_{\max}) + \log(n) + \log(c_{\max})] \,/\, \log\left(1 + \frac{\epsilon}{k^2 + k - \epsilon}\right).$$

Each iteration of the "while" loop needs to consider at most $(k|F|)^p$ different sets A and B, which is polynomial for p constant. Therefore, the time complexity of the algorithm is polynomial in the size of the input.

Note that with the new termination condition of the "while" loop we could not use inequality (2.2) to bound the locality gap, as the modified algorithm would not produce a local optimum solution, but only a solution S for which for any $A \subseteq S$ and $B \subseteq F$, $\text{cost}((S \setminus A) \cup B) > (1 - \frac{\epsilon}{|Q|})\text{cost}(S)$.

Note that if we can prove (2.2) for any local optimum solution S' then $\text{cost}(S') \leq \alpha\, \text{cost}(S^*)$. However, for the solution S obtained by the modified algorithm we know that $\text{cost}((S \setminus A_i) \cup B_i) > (1 - \frac{\epsilon}{|Q|})\text{cost}(S)$ for each $\langle A_i, B_i\rangle \in Q$; therefore, we have to modify (2.2) as follows:

$$\begin{aligned}\alpha\, \text{cost}(S^*) - \text{cost}(S) &\geq \sum_{\langle A_i, B_i\rangle \in Q} [\text{cost}((S \setminus A_i) \cup B_i) - \text{cost}(S)] \\ &> -\frac{\epsilon}{|Q|} \sum_{\langle A_i, B_i\rangle \in Q} \text{cost}(S) = -\epsilon\, \text{cost}(S).\end{aligned}$$

Hence $\text{cost}(S) \leq \frac{\alpha}{1-\epsilon}\text{cost}(S^*)$.

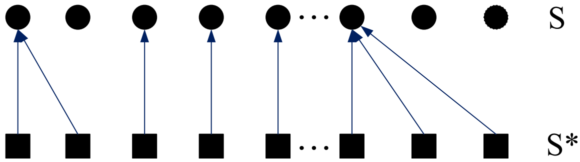

FIGURE 1. Mapping π maps each $o \in S^*$ to its closest facility $\pi(o) \in S$.

In the rest of the paper we will prove inequality (2.2) for any local optimum solution and some value α. Then, by the above argument we will have proven that our algorithm with the modified "while" condition has approximation ratio $\frac{\alpha}{1-\epsilon}$.

3. First bound for the locality gap

Since in a local optimum solution S no multi-swap operation can improve its cost, then for any multi-swap operation $\langle A, B\rangle$ the following inequality is satisfied:

$$\mathrm{cost}((S \setminus A) \cup B) \geq \mathrm{cost}(S). \tag{3.1}$$

We define a mapping π similar as the mapping η in [6] as follows: $\pi : S^* \to S$ maps each facility o in the optimum solution to its closest facility $\pi(o) \in S$ breaking ties arbitrarily; hence, $c_{o\pi(o)} \leq c_{os}$ for all $s \in S$. The map π can be represented as a bipartite graph as shown in Figure 1. For each facility $s \in S$, let the (in-)degree of s in this bipartite graph be $\deg(s) = |\pi^{-1}(s)|$. If $\deg(s) \geq 2$ then we call s a *bad facility* otherwise we call it a *good facility*.

3.1. Pairing

To bound the cost of the local optimum solution S produced by our algorithm we will use several sets of multi-swap operations involving facilities from S and facilities from S^*. These sets of multi-swap operations are chosen so that combining the local optimality condition (3.1) for all of them allows us to bound the cost of S in terms of the cost of S^*. To this end we first use algorithm Partition below to divide S and S^* into subsets of facilities that will participate in the swaps. More specifically, S and S^* are partitioned into sets $A_1, A_2, \ldots, A_r$ and $B_1, B_2, \ldots, B_r$ respectively, where $|A_i| = |B_i|$ for all $1 \leq i \leq r$ and $r-1$ is equal to the number of bad facilities.

Note the following facts:

I. In step 4 there are enough facilities s with degree 0 since $|S| = |S^*|$.
II. For any sets A_i and B_i, where $1 \leq i < r$ if $o \in S^* \setminus B_i$, then $\pi(o) \notin A_i$.
III. For each facility $s \in A_r$, $\pi^{-1}(s) = o \in B_r$. To see this note that for each facility $o \in B_r$ it must be that $\pi(o) \in A_r$ because if $\pi(o) \in A_i$ for $i \neq r$, then

Algorithm 2. Partition (S, S^*).

Input: Local optimum solution S and global optimum solution S^*
Output: Partition $(A_1, A_2, \ldots, A_r)$ of S and $(B_1, B_2, \ldots, B_r)$ of S^*
for $i \leftarrow 1$ **to** $r-1$ **do**
  $A_i \leftarrow \{b\} \cup \{\text{any } deg(b) - 1 \text{ facilities of } S \text{ with degree } 0\}$, where $b \in S$ is any bad facility
  $B_i \leftarrow \pi^{-1}(b)$
  $S \leftarrow S \setminus A_i$
  $S^* \leftarrow S^* \setminus B_i$
end for
$A_r \leftarrow S$
$B_r \leftarrow S^*$
return $(A_1, A_2, \ldots, A_r)$, $(B_1, B_2, \ldots, B_r)$

o would belong to B_i. Since $|A_r| = |B_r|$ and A_r only includes good facilities, then for all $s \in A_r$, $\deg(s) = 1$ and so $\pi^{-1}(s) \in B_r$.

We pair the facilities in S with those in S^* as follows:

- For $1 \leq i \leq r-1$ each set A_i is paired with set B_i.
- Each facility $s \in A_r$ is paired with $o \in B_r$, where $o = \pi^{-1}(s)$.

Note that if we consider multi-swaps $\langle A_i, B_i \rangle$ for $1 \leq i \leq r-1$ and single swaps $\langle s, \pi^{-1}(s) \rangle$ for all $s \in A_r$, then each facility in S and S^* would participate in one swap operation and adding all inequalities (3.1) for these swaps would allow us to bound the cost of S in terms of the cost of S^*. However, since we are only allowed to swap at most p facilities simultaneously, then the above multi-swap operations would not be allowed for those sets A_i and B_i whose size exceeds p. Therefore, we need to consider a different approach for these pairs. Algorithm Partition-2 further splits those sets A_i and B_i, where $|A_i| = |B_i| > p$, and constructs *core subsets* $\hat{A}_i \subset A_i$ and $\hat{B}_i \subset B_i$, where $|\hat{A}_i| = |\hat{B}_i| = p$; $\hat{A}_i$ includes the bad facility in A_i and $\hat{B}_i$ is built by finding a closest facility to each one of the facilities in $\hat{A}_i$.

For sets A_i and B_i, where $1 \leq i < r$ and $|A_i| = |B_i| > p$, we pair their facilities as follows:

- $\hat{A}_i$ is paired with $\hat{B}_i$.
- Each facility in $A_i \setminus \hat{A}_i$ is paired with a facility in $B_i \setminus \hat{B}_i$ so that each facility is paired once.

In the following sections, we will bound the cost of the local optimum solution S produced by our algorithm in terms of the cost of a global optimum solution S^* by considering swap operations involving the above pairs of facilities.

The idea behind the pairings. As mentioned above π maps each facility in S^* to its closest facility in S. To get some intuition as to why this is done, consider that facility $s \in S$ is close to exactly one facility $o \in S^*$ and far away from rest of facilities in S^*. Then if we swap s and o, we close facility s and open facility

Algorithm 3. Partition-2 $(A_1, A_2, \ldots, A_r, B_1, B_2, \ldots, B_r)$.

Input: Partition $(A_1, A_2, \ldots, A_r)$ of S and $(B_1, B_2, \ldots, B_r)$ of S^*
Output: Core subsets $\hat{A}_i$ and $\hat{B}_i$ for all those sets A_i and B_i for which $|A_i| = |B_i| > p$

for $i \leftarrow 1$ **to** $r-1$ **do**

    if $|A_i| > p$ **then**
        $\hat{A}_i \leftarrow \{b\} \cup \{$any $p-1$ facilities of $A_i - b\}$, where $b \in A_i$ is the bad facility in A_i
        $\hat{B}_i \leftarrow \{\}$
        $A'_i \leftarrow \hat{A}_i$

        for $i \leftarrow 1$ **to** p **do**
            $a_i \leftarrow$ any facility from A'_i
            $b_i \leftarrow$ nearest facility to a_i, where $b_i \in B_i$
            $\hat{B}_i \leftarrow \{b_i\} \cup \hat{B}_i$
            $A'_i \leftarrow A'_i \setminus \{a_i\}$
        end for
    end if
end for
return $(\hat{A}_1, \hat{A}_2, \ldots, \hat{A}_{r-1})$, $(\hat{B}_1, \hat{B}_2, \ldots, \hat{B}_{r-1})$

o reassigning the clients of s to o; this does not change the cost the solution too much, which means that the contribution of facility s to the cost of the solution is similar to the contribution of facility o to the cost of the optimum solution. However, if s is close to several facilities in S^* then performing a swap between s and the closest facility $o \in S^*$ might suggest a large difference between the cost of S and the cost of $(S \setminus \{s\}) \cup \{o\}$ because re-assigning a client j of s to o might have a much larger cost than assigning j to its closest facility in S^*. That is the reason why we call the facilities in S with degree larger than one bad and others "good" and why a "bad" facility b is not swapped in our analysis with a single facility from S^*, but instead a set of facilities containing b is swapped with a set of nearby facilities from S^*.

3.2. Analysing the swaps

Let $s_j = c_{j\sigma(j)}$ and $o_j = c_{j\sigma^*(j)}$ be the service costs of client j in solutions S and S^* respectively, where $\sigma(j)$ is the facility closest to j in S and $\sigma^*(j)$ is the facility closest to j in S^*. Let $N_S(s) = \{j \mid \sigma(j) = s\}$ be the set of clients that are served by facility s in the local optimal solution and $N^*_S(o) = \{j \mid \sigma^*(j) = o\}$ is the set of clients that are served by o in the global optimal solution. We extend these definitions to sets $A_i \subseteq S$ and $B_i \subseteq S^*$, so $N_S(A_i)$ is the set of clients that are served by facilities in A_i in S and $N^*_S(B_i)$ are those clients that are served by B_i in S^*.

All the lemmas of this section are about bounding the cost increase caused by a swap operation $\langle A, B\rangle$, where $A \subseteq S$ and $B \subseteq F \setminus S$, and they are all based on the fact that $\text{cost}((S \setminus A) \cup B) - \text{cost}(S) \geq 0$. In these lemmas we introduce a new assignment of client to facilities after the swap $\langle A, B\rangle$ is performed and bound the service cost of this assignment. More specifically, some of the clients j are re-assigned to $\pi(\sigma^*(j))$. The following lemma bounds the cost of serving client j by $\pi(\sigma^*(j))$.

Lemma 3.1 (Cost bounding)**.** $c_{j\pi(\sigma^*(j))} \leq 2o_j + s_j$.

Proof.

$$c_{j\pi(\sigma^*(j))} \leq c_{j\sigma^*(j)} + c_{\sigma^*(j)\pi(\sigma^*(j))} \tag{3.2}$$
$$\leq c_{j\sigma^*(j)} + c_{\sigma^*(j)\sigma(j)} \tag{3.3}$$
$$\leq c_{j\sigma^*(j)} + c_{j\sigma^*(j)} + c_{j\sigma(j)} = 2o_j + s_j. \tag{3.4}$$

Inequalities (6) and (8) follow from the triangle inequality and (7) is true since $\pi(\sigma^*(j))$ is the nearest facility in S to $\sigma^*(j)$ (see Fig. 2). ☐

3.2.1. Multi-swaps for sets $\mathsf{A_i}$ *and* $\mathsf{B_i}$ *where* $|\mathsf{A_i}| = |\mathsf{B_i}| \leq \mathsf{p}$

Lemma 3.2. *For each swap $\langle A_i, B_i\rangle$ where $|A_i| = |B_i| \leq p$ and $1 \leq i < r$,*

$$\sum_{o\in B_i} f_o - \sum_{s\in A_i} f_s + \sum_{j\in N_{S^*}(B_i)} (o_j - s_j) + \sum_{j\in N_S(A_i)} 2o_j \geq 0. \tag{3.5}$$

Proof. By performing swap $\langle A_i, B_i\rangle$, facilities in A_i are closed and those in B_i are opened. Therefore, the clients $j \in N_S(A_i)$ need to be re-assigned to the facilities in $(S \setminus A_i) \cup B_i$. To bound the cost of the new solution let us consider the following assignment of clients to facilities:

1. Assign all the clients in $N_S^*(B_i)$ to facilities in B_i in the same way in which they are assigned in S^*.
2. Assign each client j in $N_S(A_i) \setminus N_S^*(B_i)$ to $\hat{s} = \pi(o)$, where $o = \sigma^*(j)$ is the facility closest to j in S^*. Note that $\pi(o)$ is the closest facility to o in S and $\pi(o) \notin A_i$ by fact II in Section 3.1 (see Fig. 2).
3. The assignment of all other clients to facilities remains unchanged.

The difference in cost between solution S and $(S \setminus A_i) \cup B_i$ caused by re-assigning client $j \in N_S^*(B_i)$ to $\sigma^*(j)$ is $o_j - s_j$. Adding these cost changes over all clients in $N_S^*(B_i)$ we obtain the third term in (3.5). For clients $j \in N_S(A_i) \setminus N_S^*(B_i)$ using Lemma 3.1 the cost change is bounded by $2o_j$. Adding these changes over all clients in $N_S(A_i) \setminus N_S^*(B_i)$ gives $\sum_{j\in N_S(A_i)\setminus N_S^*(B_i)} 2o_j$. The fourth term in (3.5) is obtained by considering the fact that $\sum_{j\in N_S(A_i)\setminus N_S^*(B_i)} 2o_j \leq \sum_{j\in N_S(A_i)} 2o_j$. Finally, the first two terms in (3.5) are the result of adding the costs of the opened facilities in B_i and subtracting the costs of the closed facilities in A_i. ☐

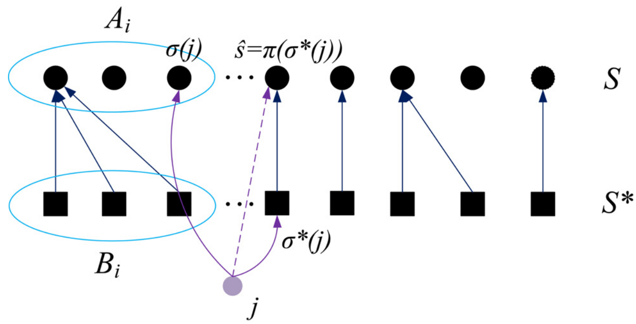

FIGURE 2. Client $j \in N_S(A_i) \backslash N^*_S(B_i)$ is assigned to $\hat{s} = \pi(\sigma^*(j))$. Note that $\sigma^*(j) \notin B_i$ and $\pi(\sigma^*(j)) \notin A_i$.

3.2.2. Swaps for sets $\mathsf{A_i}$ *and* $\mathsf{B_i}$ *where* $|\mathsf{A_i}| = |\mathsf{B_i}| > \mathsf{p}$

If $|A_i| = |B_i| = q_i > p$, we perform three different sets of swaps involving these facilities. First, we perform the swap $\langle \hat{A}_i, \hat{B}_i \rangle$.

Lemma 3.3. *For each swap* $\langle \hat{A}_i, \hat{B}_i \rangle$,

$$\sum_{o \in \hat{B}_i} f_o - \sum_{s \in \hat{A}_i} f_s + \sum_{j \in N_{S^*}(\hat{B}_i)} (o_j - s_j) + \sum_{\substack{j \in N_S(\hat{A}_i) \backslash N_{S^*}(\hat{B}_i) \\ \pi(\sigma^*(j)) \in \hat{A}_i}} (o_j + s_j) + \sum_{\substack{j \in N_S(\hat{A}_i) \backslash N_{S^*}(\hat{B}_i) \\ \pi(\sigma^*(j)) \notin \hat{A}_i}} 2o_j \geq 0. \quad (3.6)$$

Proof. Since $\hat{A}_i$ and $\hat{B}_i$ are subsets of A_i and B_i respectively, then fact II in Section 3.1 might not hold for them. In other words for some clients $j \in N_S(\hat{A}_i) \setminus N^*_S(\hat{B}_i)$ it might be that $\pi(\sigma^*(j)) \in \hat{A}_i$. Therefore, if we want to proceed similarly as in the proof of Lemma 3.2 we need to define a new re-assignment for clients $j \in N_S(\hat{A}_i) \setminus N^*_S(\hat{B}_i)$ for which $\pi(\sigma^*(j)) \in \hat{A}_i$. Consider the following assignment of clients to facilities in $(S \setminus \hat{A}_i) \cup \hat{B}_i$:

1. Assign client $j \in N^*_S(\hat{B}_i)$ to $\sigma^*(j) \in \hat{B}_i$; this changes the cost of S by $o_j - s_j$. Adding these cost changes over all clients in $N^*_S(\hat{B}_i)$ gives the third term in (3.6).
2. Assign each client j in $N_S(\hat{A}_i) \setminus N^*_S(\hat{B}_i)$ such that $\pi(\sigma^*(j)) \notin \hat{A}_i$ to $\pi(\sigma^*(j))$. Using Lemma 3.1 the cost increase per client j for this reassignment is $2o_j$. Adding these cost increases over all these clients gives us the fifth term in (3.6).
3. Consider a client $j \in N_S(\hat{A}_i) \setminus N^*_S(\hat{B}_i)$ for which $\pi(\sigma^*(j)) \in \hat{A}_i$. Let $s \in \hat{A}_i$ be the facility serving j in S and $o \in \hat{B}_i$ be the closest facility to s; client j is

assigned to o. Let o' be the facility serving j in S^*. The change in cost caused by reassigning client j to o is $c_{jo} - s_j$. Note that

$$\begin{aligned} c_{jo} - s_j &\leq c_{js} + c_{so} - s_j && (3.7)\\ &\leq c_{js} + c_{so'} - s_j && (3.8)\\ &\leq c_{js} + c_{js} + c_{jo'} - s_j = o_j + s_j. && (3.9)\end{aligned}$$

Inequalities (3.7) and (3.9) hold because of the triangle inequality and (3.8) is true because o is closer to s than o'. Adding these cost increases over all client in $N_S(\hat{A}_i) \setminus N^*_S(\hat{B}_i)$ for which $\pi(\sigma^*(j)) \in (\hat{A}_i)$ gives us the fourth term in (3.6).

4. The assignment of the rest of the clients to facilities remains unchanged.

Finally, the first two terms in (3.6) are the result of adding the costs of all the opened facilities in $\hat{B}_i$ and subtracting the cost of the closed ones in $\hat{A}_i$. □

Note that the fourth term in inequality (3.6) includes the service cost s_j for some clients as a positive term. Since our goal is to find an upper bound for the cost of the local optimum solution S, the appearance of these positive service costs s_j on the left side of (3.6) is problematic. To get rid of these terms we perform a second set of swaps for pairs $\langle s, o\rangle$, where $s \in A_i \setminus \hat{A}_i$ and $o \in B_i \setminus \hat{B}_i$.

Corollary 3.4. *For each swap $\langle s, o\rangle$ where $s \in A_i \setminus \hat{A}_i$ and $o \in B_i \setminus \hat{B}_i$,*

$$f_o - f_s + \sum_{\substack{j \in N_{S^*}(o) \cap N_S(\hat{A}_i),\\ \pi(\sigma^*(j)) \in \hat{A}_i}} (o_j - s_j) + \sum_{j \in N_S(s)} 2o_j \geq 0. \tag{3.10}$$

Proof. In Lemma 3.2 replace A_i with s and B_i with o and change the first and second re-assignment of clients to facilities as follows:

1. Assign all clients $j \in N_{S^*}(o) \cap N_S(\hat{A}_i)$ such that $\pi(\sigma^*(j)) \in \hat{A}_i$ to o.
2. Assign all clients $j \in N_S(s)$ to $\pi(\sigma^*(j))$. Note that this is a valid assignment because s is a good facility with degree 0; therefore, $\pi(o) \neq s$ for every facility $o \in S^*$. □

Lemma 3.5. *For each $i = 1, \ldots, r-1$ such that $|A_i| = |B_i| > p$,*

$$\sum_{o \in B_i \setminus \hat{B}_i} f_o - \sum_{s \in A_i \setminus \hat{A}_i} f_s + \sum_{\substack{j \in N_S(\hat{A}_i) \setminus N^*_S(\hat{B}_i),\\ \pi(\sigma^*(j)) \in \hat{A}_i}} (o_j - s_j) + \sum_{j \in N_S(A_i) \setminus N_S(\hat{A}_i)} 2o_j \geq 0. \tag{3.11}$$

Proof. By fact II in Section 3.1, for any sets A_i and B_i, where $1 \leq i < r$, if $o \in S^* \setminus B_i$ then $\pi(o) \notin A_i$; thus, if $\pi(o) \in A_i$ then $o \in B_i$. Consequently, for a client $j \in N_S(\hat{A}_i)$ if $\pi(\sigma^*(j)) \in \hat{A}_i$ then j belongs to $N^*_S(B_i)$; therefore, $\{j \mid j \in N_S(\hat{A}_i), \pi(\sigma^*(j)) \in \hat{A}_i\} \subseteq N^*_S(B_i)$ and so $\{j \mid j \in N^*_S(B_i) \cap N_S(\hat{A}_i), \pi(\sigma^*(j)) \in \hat{A}_i\} = \{j \mid j \in N_S(\hat{A}_i), \pi(\sigma^*(j)) \in \hat{A}_i\}$. Hence,

$$\begin{aligned}&\{j \mid j \in [N^*_S(B_i) \setminus N^*_S(\hat{B}_i)] \cap N_S(\hat{A}_i), \pi(\sigma^*(j)) \in \hat{A}_i\} = \\ &\{j \mid j \in [N^*_S(B_i) \cap N_S(\hat{A}_i)] \setminus N^*_S(\hat{B}_i), \pi(\sigma^*(j)) \in \hat{A}_i\} = \{j \mid j \in N_S(\hat{A}_i) \setminus N^*_S(\hat{B}_i), \\ &\qquad \pi(\sigma^*(j)) \in \hat{A}_i\}. \end{aligned} \tag{3.12}$$

Adding inequality (3.10) over all facilities $o \in B_i \setminus \hat{B}_i$ and $s \in A_i \setminus \hat{A}_i$ and using (3.12) we get (3.11). □

As mentioned before since we do not want the positive service cost s_j in the left side of (3.6), we add (3.6) and (3.11) to discard the undesired terms

$$\sum_{o\in B_i} f_o - \sum_{s\in A_i} f_s + \sum_{j\in N_{S^*}(\hat{B}_i)} (o_j - s_j) + \sum_{j\in N_S(A_i)} 2o_j \geq 0. \tag{3.13}$$

To get the fourth term in (3.13) note that adding the third term in (3.11) and the fourth term in (3.6) we get

$$\sum_{\substack{j\in N_S(\hat{A}_i)\setminus N^*_S(\hat{B}_i),\\ \pi(\sigma^*(j))\in\hat{A}_i}} (o_j - s_j) + \sum_{\substack{j\in N_S(\hat{A}_i)\setminus N^*_S(\hat{B}_i),\\ \pi(\sigma^*(j))\in\hat{A}_i}} (o_j + s_j) = \sum_{\substack{j\in N_S(\hat{A}_i)\setminus N^*_S(\hat{B}_i),\\ \pi(\sigma^*(j))\in\hat{A}_i}} 2o_j.$$

Adding the right hand side of the above equality and the fifth term in (3.6) yields

$$\sum_{\substack{j\in N_S(\hat{A}_i)\setminus N^*_S(\hat{B}_i),\\ \pi(\sigma^*(j))\in\hat{A}_i}} 2o_j + \sum_{\substack{j\in N_S(\hat{A}_i)\setminus N^*_S(\hat{B}_i),\\ \pi(\sigma^*(j))\notin\hat{A}_i}} 2o_j = \sum_{j\in N_S(\hat{A}_i)\setminus N^*_S(\hat{B}_i)} 2o_j.$$

Finally, adding the right hand side of the above equality and the fourth term in (3.11) we get

$$\sum_{j\in N_S(\hat{A}_i)\setminus N^*_S(\hat{B}_i)} 2o_j + \sum_{j\in N_S(A_i)\setminus N_S(\hat{A}_i)} 2o_j \leq \sum_{j\in N_S(A_i)} 2o_j.$$

As our goal is to find an upper bound for $\mathrm{cost}(S)$, we note that the left hand side of inequality (3.13) is missing the service cost of the clients that are served by facilities in $B_i \setminus \hat{B}_i$. To include this missing cost we perform a third set of swaps, where each good facility in A_i is swapped with every facility in $B_i \setminus \hat{B}_i$.

Corollary 3.6. *For each swap $\langle s, o\rangle$ where s is a good facility in A_i and $o \in B_i$,*

$$f_o - f_s + \sum_{j\in N_{S^*}(o)} (o_j - s_j) + \sum_{j\in N_S(s)} 2o_j \geq 0. \tag{3.14}$$

Proof. In Lemma 3.2 replace A_i with s and B_i with o and note that assigning all clients $j \in N_S(s) \setminus N_{S^*}(o)$ to $\pi(\sigma^*(j))$ is a valid assignment because s is a good facility and so $s \neq \pi(o)$ for all $o \in S^*$. □

Let b_i be the bad facility in A_i and let $q_i = |A_i| = |B_i|$. Adding the inequality (3.14) for all pairs $\langle s, o\rangle \in (A_i - b_i) \times (B_i \setminus \hat{B}_i)$, we get

$$(q_i - 1) \sum_{o \in B_i \setminus \hat{B}_i} f_o - (q_i - p) \sum_{s \in A_i - b_i} f_s + (q_i - 1) \sum_{j \in N_S^*(B_i) \setminus N_S^*(\hat{B}_i)} (o_j - s_j) \\ + (q_i - p) \sum_{j \in N_S(A_i - b_i)} 2o_j \geq 0. \quad (3.15)$$

Observe that each facility $o \in B_i$ is swapped $q_i - 1$ times, therefore facility cost f_o and service cost change $o_j - s_j$ for clients $j \in N_{S^*}(B_i) \setminus N_{S^*}(\hat{B}_i)$ are added $q_i - 1$ times. In addition, each good facility $s \in A_i$ is swapped $q_i - p$ times, therefore facility cost $-f_s$ and service cost $2o_j$ (the fourth term in (3.14)) for clients $j \in N_S(s)$ are added $q_i - p$ times.

Multiplying (3.15) by $\frac{1}{q_i - 1}$ and adding to (3.13) we get

$$\sum_{o \in B_i} f_o + \sum_{o \in B_i \setminus \hat{B}_i} f_o - \sum_{s \in A_i} f_s - \frac{q_i - p}{q_i - 1} \sum_{s \in A_i - b_i} f_s + \sum_{j \in N_{S^*}(B_i)} (o_j - s_j) + \sum_{j \in N_S(A_i)} 2o_j \\ + \frac{q_i - p}{q_i - 1} \sum_{j \in N_S(A_i - b_i)} 2o_j \geq 0. \quad (3.16)$$

Using $\sum_{o \in B_i \setminus B_i^*} f_o \leq \sum_{o \in B_i} f_o$, $\frac{q_i - p}{q_i - 1} \sum_{s \in A_i - b_i} f_s > 0$ and $\sum_{j \in N_S(b_i)} 2o_j > 0$ in (3.16) we get

$$2 \sum_{o \in B_i} f_o - \sum_{s \in A_i} f_s + \sum_{j \in N_{S^*}(B_i)} (o_j - s_j) + \left(2 - \frac{p - 1}{q_i - 1}\right) \sum_{j \in N_S(A_i)} 2o_j \geq 0. \quad (3.17)$$

3.2.3. Single swaps for facilities in sets $\mathsf{A_r}$ *and* $\mathsf{B_r}$

Corollary 3.7. *For each swap $\langle s, o\rangle$, where $s \in A_r$ has been paired with $o \in B_r$,*

$$f_o - f_s + \sum_{j \in N_{S^*}(o)} (o_j - s_j) + \sum_{j \in N_S(s)} 2o_j \geq 0. \quad (3.18)$$

Proof. In Lemma 3.2 replace A_i with s and B_i with o and note that assigning all clients $j \in N_S(s) \setminus N_{S^*}(o)$ to $\pi(\sigma^*(j))$ is a valid assignment since if $j \notin N_S(o)$ then $\pi(\sigma^*(j)) \neq s$ as $s = \pi(o)$ for each pair (s, o) in sets A_r and B_r. $\square$

Adding the inequalities (3.18) for all pairs (s, o) in sets A_r and B_r we get

$$\sum_{o \in B_r} f_o - \sum_{s \in A_r} f_s + \sum_{j \in N_{S^*}(B_r)} (o_j - s_j) + \sum_{j \in N_S(A_r)} 2o_j \geq 0. \quad (3.19)$$

3.2.4. Putting it all together

Let $G \subseteq S$ be the set of facilities in all sets A_i, $1 \leq i < r$ for which $|A_i| > p$. These facilities are swapped with some facilities in S^* as explained in Section 3.2.2; let $G^* \subseteq S^*$ be this set of facilities. Let $I = \{i \mid 1 \leq i < r, |A_i| > p\}$ and $I^c = \{i \mid 1 \leq i < r, |A_i| \leq p\}$. Adding inequalities (3.5) for all sets A_i, $i \in I^c$, inequalities (3.17) for all sets A_i, $i \in I$, and inequality (3.19) we get

$$\sum_{o \in S^*} f_o + \sum_{o \in G^*} f_o - \sum_{s \in S} f_s + \sum_{j \in C} (o_j - s_j) + \sum_{j \in C \setminus N_S(G)} 2o_j + \sum_{i \in I} \left[\left(2 - \frac{p-1}{q_i - 1} \right) \sum_{j \in N_S(A_i)} 2o_j \right] \geq 0. \quad (3.20)$$

Lemma 3.8.

$$\sum_{i \in I} \left[\left(2 - \frac{p-1}{q_i - 1} \right) \sum_{j \in N_S(A_i)} 2o_j \right] \leq \left(2 - \frac{p-1}{q-1} \right) \sum_{j \in N_S(G)} 2o_j$$

where $q = \max \{q_i | \ i \in I\}$.

Proof. The lemma follows since $2o_j$ is positive. □

Using Lemma 3.8 inequality (3.20) can be rewritten as follows:

$$\sum_{o \in S^*} f_o + \sum_{o \in G^*} f_o - \sum_{s \in S} f_s + \sum_{j \in C} (o_j - s_j) + \sum_{j \in C} 2o_j + \left(1 - \frac{p-1}{q-1} \right) \sum_{j \in N_S(G)} 2o_j \geq 0. \quad (3.21)$$

The fifth term in (3.21) is obtained by adding $\sum_{j \in N_S(G)} 2o_j$ to the fifth term of (3.20) and the sixth term in (3.21) is obtained by subtracting $\sum_{j \in N_S(G)} 2o_j$ from the last term of (3.20).

Since $\sum_{j \in N_S(G)} 2o_j \leq \sum_{j \in C} 2o_j$, $G \subseteq S$, and all the facility and service costs are positive then

$$\begin{aligned} 0 &\leq 2 \sum_{o \in S^*} f_o - \sum_{s \in S} f_s + \sum_{j \in C} (o_j - s_j) + \left(2 - \frac{p-1}{q-1} \right) \sum_{j \in C} 2o_j \\ &= 2\mathrm{cost}_f(S^*) + \left[5 - 2 \left(\frac{p-1}{q-1} \right) \right] \mathrm{cost}_s(S^*) - \mathrm{cost}_f(S) - \mathrm{cost}_s(S). \end{aligned}$$

Therefore, since $p < q$ and so $\left[5 - 2 \left(\frac{p-1}{q-1} \right) \right] > 2$, then

$$\left[5 - 2 \left(\frac{p-1}{q-1} \right) \right] (\mathrm{cost}_f(S^*) + \mathrm{cost}_s(S^*)) \geq \mathrm{cost}_f(S) + \mathrm{cost}_s(S).$$

Notice that if no set A_i has size larger than p then the swaps considered in Section 3.2.3 are not needed and in that case it can be shown that

$$3(\mathrm{cost}_f(S^*) + \mathrm{cost}_s(S^*)) \geq \mathrm{cost}_f(S) + \mathrm{cost}_s(S).$$

Theorem 3.9. *The locality gap of the local search algorithm where the only operation allowed is multiple swaps is* $\max\{3, 5 - 2\left(\frac{p-1}{q-1}\right)\}$, *where q is the size of the largest set A_i.*

The total number of multi-swap operations considered in Lemmas 3.2 and 3.3 and Corollaries 3.6 and 3.7 is at most k. The number of multi-swap operations considered in Corollary 3.7 is at most k^2. Therefore, the number of multi-swap operations considered by our analysis is at most $k^2 + k$.

3.3. Special case when the ratio of the biggest facility cost to the smallest facility cost is less than $\mathsf{p+1}$

Add inequality (3.14) for all pairs $\langle s, o\rangle \in (A_i - b_i) \times B_i$ and then multiply by $\frac{1}{q_i - 1}$ to get

$$\sum_{o \in B_i} f_o - \frac{q_i}{q_i - 1} \sum_{s \in A_i - b_i} f_s + \sum_{j \in N_S^*(B_i)} (o_j - s_j) + \frac{q_i}{q_i - 1} \sum_{j \in N_S(A_i - b_i)} 2o_j \geq 0. \quad (3.22)$$

Since the ratio of the biggest facility cost to the smallest facility cost is less than $p+1$ then $\sum_{o \in B_i} f_o - f_{b_i} \geq 0$. Also, since $q_i \geq p+1$ then $\frac{p+1}{p} \geq \frac{q_i}{q_i - 1}$. Therefore, if we add the following non-negative terms $\sum_{o \in B_i} f_o - f_{b_i}$, $\frac{1}{q_i-1} \sum_{s \in A_i - b_i} f_s$, $\frac{q_i}{q_i - 1} \sum_{j \in N_S(b_i)} 2o_j$ to (3.22) and replace $\frac{q_i}{q_i-1}$ with $\frac{p+1}{p}$ we get

$$2 \sum_{o \in B_i} f_o - \sum_{s \in A_i} f_s + \sum_{j \in N_{S^*}(B_i)} (o_j - s_j) + \frac{p+1}{p} \sum_{j \in N_S(A_i)} 2o_j \geq 0. \quad (3.23)$$

If we proceed similarly as in Section 3.2.4 and add inequalities (3.5) for all sets A_i, $i \in I^c$, inequalities (3.23) for all sets A_i, $i \in I$, and inequality (3.19), we get

$$\left[3 + \frac{2}{p}\right] (\mathrm{cost}_f(S^*) + \mathrm{cost}_s(S^*)) \geq \mathrm{cost}_f(S) + \mathrm{cost}_s(S).$$

Theorem 3.10. *The locality gap of the local search algorithm where the only operation allowed is multiple swaps for the special case when the ratio of the biggest facility cost to the smallest facility cost is less than $p+1$ is* $3 + \frac{2}{p}$.

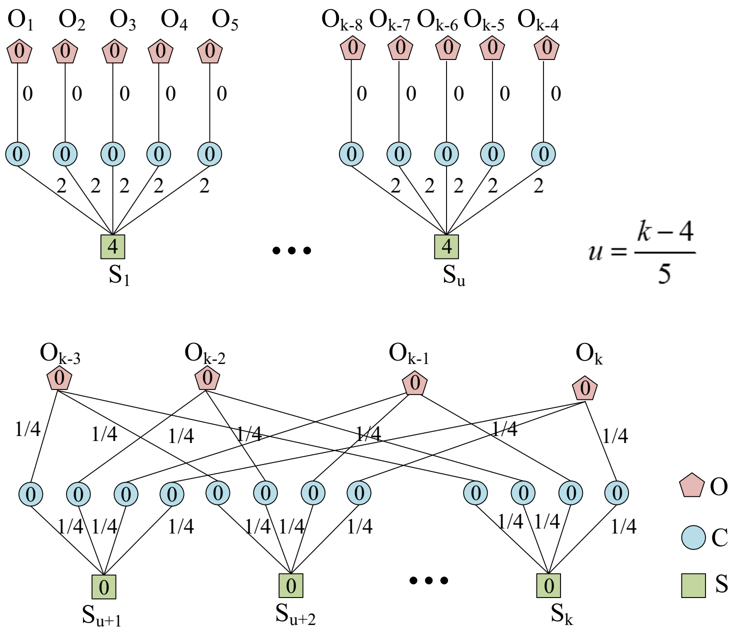

FIGURE 3. Tight example for $p = 2$ and $q = 5$.

3.4. Tight example

Figure 3 illustrates an instance of the k-facility location problem showing a locality gap of 4.5 for $p = 2$ and $q = 5$, matching the locality gap stated in Theorem 3.9. In Figure 3 the pentagonal nodes form the global optimal solution $S^* = \{o_1, o_2, \ldots, o_k\}$, the square nodes are the local optimal solution $S = \{s_1, s_2, \ldots, s_k\}$, and the circular nodes are the clients. The facility costs are the integers inside the nodes and the service costs are equal to the lengths of the shortest paths between the corresponding nodes. The length of each edge is shown beside the edge and if there is no path between two nodes the distance between them is infinity.

In the instance shown in Figure 3, cost$(S) = \frac{18k-52}{5}$ and cost$(S^*) = \frac{4k+4}{5}$, therefore the locality gap is $\frac{18k-52}{4k+4}$ which approaches 4.5 as k grows. We now prove that S is locally optimal by considering all possible swaps. Let set $\{s_i, s_j\} \subset S$ be swapped with $\{o_l, o_m\} \subset S^*$.

1. If $i, j \leq u = \frac{k-4}{5}$, then o_l and o_m should lie in the same connected components containing s_i and s_j, so this swap increases the cost by 4.
2. If $i \leq u < j$, then one of o_l, o_m should lie in the same connected component as s_i. Without loss of generality, consider that o_l lies in the same component as s_i. If o_m lies in the same component also, the cost remains unchanged. If $m \leq k-4$

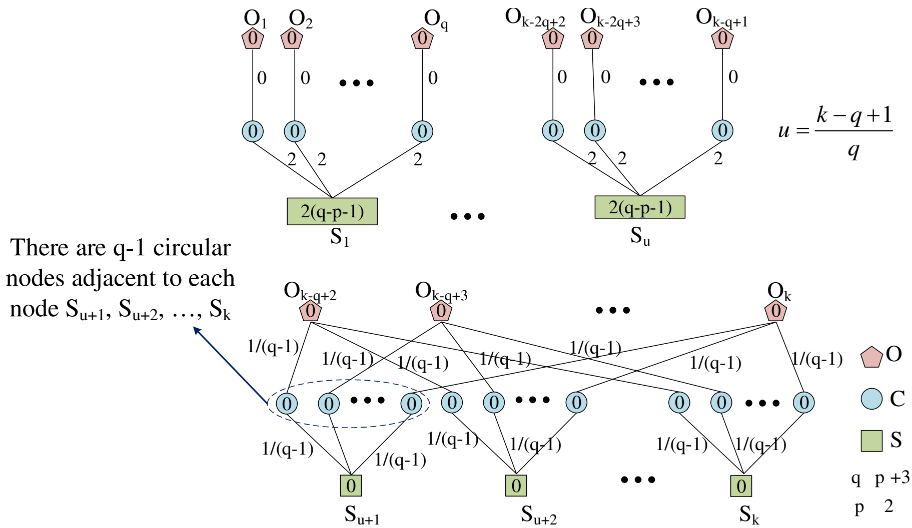

FIGURE 4. Tight example for arbitrary p and q.

and o_m lies in a different component than s_i the cost increases by 2. If $m > k-4$ the cost increases by 3.5.

3. If $i, j > u$, there are three cases that need to be considered. First, if $l, m \leq k-4$ the cost remains unchanged. Second, if $l \leq k-4 < m$ then the cost increases by 1. Third, if $k-4 < l, m$ the cost increases by 2.

The example can be generalized to arbitrary values of $p \geq 2$ and $q \geq p+3$ as shown in Figure 4. The local optimal solution is $S = \{s_1, s_2, \ldots, s_k\}$ and the global optimal solution is $S^* = \{o_1, o_2, \ldots, o_k\}$. The cost of S is $\frac{[5(q-1)-2(p-1)]k-[(q-1)(4q-2p-3)]}{q}$ and the cost of S^* is $\frac{(q-1)k+(q-1)}{q}$, so the locality gap is $\frac{5(q-1)-2(p-1)}{q-1} = 5 - 2\left(\frac{p-1}{q-1}\right)$. Note that in our tight example k must be much larger than q.

The proof that S is locally optimal is similar as that for the case $p = 2$ and $q = 5$, but it involves many more cases.

4. Scaling the costs

We now perform a different analysis of our algorithm which yields the same bound for the locality ratio as that of the algorithm by Zhang [13] which uses facility insertions and removals in addition to swaps. The idea is to multiply each facility cost by some value $\beta > 0$ and then compute a local optimal solution for the new problem. By carefully choosing the value of β and some set of swap operations involving facilities from the local optimum solution and a global optimum solution we can prove a locality ratio that matches that of [13].

4.1. Bounding the facility cost

This time we consider the following swap operations:

1. Swap $\langle A_i, B_i\rangle$, where $|A_i| \le p$.
2. Swap $\langle \hat{A}_i, \hat{B}_i\rangle$, where $|A_i| > p$, and swaps $\langle\langle A_i \setminus \hat{A}_i, B_i \setminus \hat{B}_i\rangle\rangle$, where $\langle\langle A_i \setminus \hat{A}_i, B_i \setminus \hat{B}_i\rangle\rangle$ denotes the set of single swaps for each pair of facilities in $A_i \setminus \hat{A}_i$ and $B_i \setminus \hat{B}_i$.
3. Swaps $\langle\langle A_r, B_r\rangle\rangle$, where $\langle\langle A_r, B_r\rangle\rangle$ denotes a set of single swaps for each pair of facilities in A_r and B_r.

Corollary 4.1. *For swap $\langle A_i, B_i\rangle$, where $|A_i| \le p$ and $1 \le i < r$,*

$$\sum_{o\in B_i} f_o - \sum_{s\in A_i} f_s + \sum_{j\in N_S(A_i)} 2o_j \ge 0. \tag{4.1}$$

Proof. The corollary follows from Lemma 3.2. Note that as pointed out at the end of the proof of Lemma 3.2 we can replace in (3.5) the last term $\sum_{j\in N_S(A_i)} 2o_j$ with $\sum_{j\in N_S(A_i)\setminus N_S(B_i)} 2o_j$. In addition, since $o_j - s_j \le 2o_j$ the third term in (3.5) can be replaced by $\sum_{N_S(B_i)} 2o_j$. The third term in (4.1) is then obtained by adding $\sum_{j\in N_S(A_i)\setminus N_S(B_i)} 2o_j$ to $\sum_{N_S(B_i)} 2o_j$. $\square$

Corollary 4.2. *For swap $\langle \hat{A}_i, \hat{B}_i\rangle$, where $|A_i| > p$ and $1 \le i < r$,*

$$\sum_{o\in \hat{B}_i} f_o - \sum_{s\in \hat{A}_i} f_s + \sum_{j\in N_S^*(\hat{B}_i)} 2o_j + \sum_{\substack{j\in N_S(\hat{A}_i)\setminus N_{S^*}(\hat{B}_i)\\ \pi(\sigma^*(j))\in \hat{A}_i}} (o_j + s_j) + \sum_{\substack{j\in N_S(\hat{A}_i)\setminus N_{S^*}(\hat{B}_i)\\ \pi(\sigma^*(j))\notin \hat{A}_i}} 2o_j \ge 0. \tag{4.2}$$

Proof. The corollary follows from Lemma 3.3 by noting that $o_j - s_j \le 2o_j$ for all clients j, and using this inequality in the third term of (3.6). $\square$

Lemma 3.5 bounds the cost increase caused by swaps $\langle s, o\rangle$, where $s \in A_i \setminus \hat{A}_i$, and $o \in B_i \setminus \hat{B}_i$. Adding (4.2) and (3.11) we get

$$\sum_{o\in B_i} f_o - \sum_{s\in A_i} f_s + \sum_{j\in N_S(A_i)} 2o_j \ge 0. \tag{4.3}$$

The third term in (4.3) is obtained from the following equalities: adding the third term of (3.11) and the fourth term in (4.2) we get

$$\sum_{\substack{j\in N_S(\hat{A}_i)\setminus N_S^*(\hat{B}_i),\\ \pi(\sigma^*(j))\in \hat{A}_i}} (o_j - s_j) + \sum_{\substack{j\in N_S(\hat{A}_i)\setminus N_S^*(\hat{B}_i),\\ \pi(\sigma^*(j))\in \hat{A}_i}} (o_j + s_j) = \sum_{\substack{j\in N_S(\hat{A}_i)\setminus N_S^*(\hat{B}_i),\\ \pi(\sigma^*(j))\in \hat{A}_i}} 2o_j.$$

Adding the right hand side of this equation and the last term in (4.2) we have

$$\sum_{\substack{j\in N_S(\hat{A}_i)\setminus N_S^*(\hat{B}_i),\\ \pi(\sigma^*(j))\in\hat{A}_i}} 2o_j + \sum_{\substack{j\in N_S(\hat{A}_i)\setminus N_S^*(\hat{B}_i),\\ \pi(\sigma^*(j))\notin\hat{A}_i}} 2o_j = \sum_{j\in N_S(\hat{A}_i)\setminus N_S^*(\hat{B}_i)} 2o_j.$$

Finally, adding the right hand side of this equation to the third term in (4.2) and the last term in (3.11) we get

$$\sum_{j\in N_S(\hat{A}_i)\setminus N_S^*(\hat{B}_i)} 2o_j + \sum_{j\in N_S^*(\hat{B}_i)} 2o_j + \sum_{j\in N_S(A_i)\setminus N_S(\hat{A}_i)} 2o_j = \sum_{j\in N_S(A_i)} 2o_j.$$

In Corollary 4.1 if we replace A_i with s and B_i with o, then for each swap $\langle s,o\rangle$, where $s\in A_r$ and $o\in B_r$, we have

$$f_o - f_s + \sum_{j\in N_S(s)} 2o_j \geq 0. \tag{4.4}$$

Adding all the cost increase inequalities for swaps $\langle s,o\rangle$, where $s\in A_r$ and $o\in B_r$, we get

$$\sum_{o\in B_r} f_o - \sum_{s\in A_r} f_s + \sum_{j\in N_S(A_r)} 2o_j \geq 0. \tag{4.5}$$

Adding inequalities (4.1) for all subsets A_i, where $|A_i|\leq p$, inequalities (4.3) for all subsets A_i, where $|A_i|>p$, and inequality (4.5) we get

$$\sum_{o\in S^*} f_o - \sum_{s\in S} f_s + 2\sum_{j\in C} o_j \geq 0. \tag{4.6}$$

Therefore,

$$\mathrm{cost}_f(S) \leq \mathrm{cost}_f(S^*) + 2\mathrm{cost}_s(S^*). \tag{4.7}$$

4.2. Bounding the service cost

For bounding the service cost of the local optimum solution S we consider the following swaps:

1. Swap $\langle A_i, B_i\rangle$, where $|A_i|\leq p$.
2. For each set A_i with $q_i=|A_i|>p$, $1\leq i<r$, we pair each of the q_i-1 good facilities in A_i with all q_i facilities in B_i; this produces $(q_i-1)q_i$ different pairs. For set A_r we select any $q_r-1=|A_r|-1$ good facilities in A_r and we pair each one of them with all facilities in B_r. For each $i=1,2,\ldots,r$, we swap each one of the $(q_i-1)q_i$ pairs of facilities.

By Lemma 3.2, for each swap $\langle A_i, B_i\rangle$, where $|A_i|\leq p$,

$$\sum_{o\in B_i} f_o - \sum_{s\in A_i} f_s + \sum_{j\in N_{S^*}(B_i)} (o_j - s_j) + \sum_{j\in N_S(A_i)} 2o_j \geq 0. \tag{4.8}$$

By Lemma 3.6, for each swap $\langle s, o\rangle$, where $s \in A_i$ is a good facility, $o \in B_i$, $1 \le i < r$ and $|A_i| > p$,

$$f_o - f_s + \sum_{j\in N_S^*(o)} (o_j - s_j) + \sum_{j\in N_S(s)} 2o_j \ge 0. \tag{4.9}$$

Adding all the cost increase inequalities related to swapping all $q_i(q_i-1)$ pairs $\langle s, o\rangle$, where $s \in A_i$ is a good facility and $o \in B_i$, and then multiplying by $\frac{1}{q_i-1}$ we get

$$\sum_{o\in B_i} f_o - \frac{q_i}{q_i-1}\sum_{s\in A_i\setminus b_i} f_s + \sum_{j\in N_{S^*}(B_i)} (o_j - s_j) + \frac{q_i}{q_i-1}\sum_{j\in N_S(A_i)} 2o_j \ge 0, \tag{4.10}$$

where $b_i \in A_i$ is a bad facility.

Since $\frac{q_i}{q_i-1} \le \frac{p+1}{p}$ and $2o_j \ge 0$, inequality (4.10) can be rewritten as

$$\sum_{o\in B_i} f_o - \frac{q_i}{q_i-1}\sum_{s\in A_i\setminus b_i} f_s + \sum_{j\in N_{S^*}(B_i)} (o_j - s_j) + \frac{p+1}{p}\sum_{j\in N_S(A_i)} 2o_j \ge 0. \tag{4.11}$$

Let $I = \{i \mid 1 \le i \le r, |A_i| > p\}$ and $I^c = \{i \mid 1 \le i \le r, |A_i| \le p\}$. Adding inequalities (4.8) for all sets A_i, where $i \in I^c$, and inequalities (4.11) for all A_i, where $i \in I$, we get

$$\sum_{o\in S^*} f_o - \sum_{i\in I^c}\sum_{s\in A_i} f_s - \sum_{i\in I}\left[\left(\frac{q_i}{q_i-1}\right)\sum_{s\in A_i\setminus b_i} f_s\right] + \sum_{j\in C} o_j - \sum_{j\in C} s_j + \left(\frac{p+1}{p}\right)\sum_{j\in C} 2o_j \ge 0. \tag{4.12}$$

The sixth term is obtained by noting that $\sum_{i\in I^c}\left(\frac{p+1}{p}\right)\sum_{j\in N_S(A_i)} 2o_j \ge \sum_{i\in I^c}\sum_{j\in N_S(A_i)} 2o_j$.

Therefore,

$$\mathrm{cost}_f(S^*) - \sum_{i\in I^c}\sum_{s\in A_i} f_s - \sum_{i\in I}\left[\left(\frac{q_i}{q_i-1}\right)\sum_{s\in A_i\setminus b_i} f_s\right] + \mathrm{cost}_s(S^*) - \mathrm{cost}_s(S) + \left(\frac{p+1}{p}\right) 2\mathrm{cost}_s(S^*) \ge 0$$

$$\Rightarrow \sum_{i\in I^c}\sum_{s\in A_i} f_s + \sum_{i\in I}\left[\left(\frac{q_i}{q_i-1}\right)\sum_{s\in A_i\setminus b_i} f_s\right] + \mathrm{cost}_s(S) \le \mathrm{cost}_f(S^*) + \left(3+\frac{2}{p}\right)\mathrm{cost}_s(S^*). \tag{4.13}$$

Since the term $\sum_{i \in I^c} \sum_{s \in A_i} f_s + \sum_{i \in I} \left[\left(\frac{q_i}{q_i - 1} \right) \sum_{s \in A_i \setminus b_i} f_s \right]$ is positive, we can omit it from inequality (4.13) and we get

$$\mathrm{cost}_s(S) \leq \mathrm{cost}_f(S^*) + \left[3 + \frac{2}{p} \right] \mathrm{cost}_s(S^*). \tag{4.14}$$

Theorem 4.3. *Algorithm local search has locality gap* $2 + \frac{1}{p} + \sqrt{3 + \frac{2}{p} + \frac{1}{p^2}}$.

Proof. Consider an instance (F, C) of the k-facility location problem. Multiply the cost of each facility in F by some value $\beta > 0$ and compute a local optimum solution S for the new instance. Let $\mathrm{cost}'_f(X)$ and $\mathrm{cost}'_s(X)$ denote respectively the facility and service cost of solution X in the new scaled problem, and let $\mathrm{cost}_f(X)$ and $\mathrm{cost}_s(X)$ be the facility and service cost of solution X with the original costs. Using inequalities (4.7) and (4.14) we get

$$\mathrm{cost}'_f(S) \leq \mathrm{cost}'_f(S^*) + 2\mathrm{cost}'_s(S^*)$$

and

$$\mathrm{cost}'_s(S) \leq \mathrm{cost}'_f(S^*) + \left(3 + \frac{2}{p} \right) \mathrm{cost}'_s(S^*).$$

Since

$$\mathrm{cost}_f(S) + \mathrm{cost}_s(S) = \frac{\mathrm{cost}'_f(S)}{\beta} + \mathrm{cost}'_s(S)$$

then

$$\begin{aligned}
\mathrm{cost}_f(S) + \mathrm{cost}_s(S) &\leq \frac{\mathrm{cost}'_f(S^*) + 2\mathrm{cost}'_s(S^*)}{\beta} + \mathrm{cost}'_f(S^*) + \left(3 + \frac{2}{p} \right) \mathrm{cost}'_s(S^*) \\
&= \left(1 + \frac{1}{\beta} \right) \mathrm{cost}'_f(S^*) + \left(3 + \frac{2}{p} + \frac{2}{\beta} \right) \mathrm{cost}'_s(S^*) \\
&= (\beta + 1)\mathrm{cost}_f(S^*) + \left(3 + \frac{2}{p} + \frac{2}{\beta} \right) \mathrm{cost}_s(S^*).
\end{aligned}$$

By setting $\beta = 1 + \frac{1}{p} + \sqrt{3 + \frac{2}{p} + \frac{1}{p^2}}$ we get

$$\mathrm{cost}(S) \leq \left[2 + \frac{1}{p} + \sqrt{3 + \frac{2}{p} + \frac{1}{p^2}} \right] \mathrm{cost}(S^*). \qquad \square$$

The total number of multi-swap operations considered in Sections 4.1 and 4.2 is at most $k^2 + k$.

References

[1] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala and V. Pandit, Local search heuristic for k-median and facility location problem. *SIAM J. Comput.* **33** (2004) 544–562.

[2] N. Atay and B. Bayazit, Mobile Wireless Sensor Network Connectivity Repair with K-Redundancy. Algorithmic Foundations of Robotics, edited by G.S. Chirikjian, H. Choset, M. Morales and T. Murphey. *Springer Tracts Adv. Rob.* (2010) 35–52.

[3] M. Charikar and S. Guha, Improved Combinatorial Algorithms for Facility Location Problems. *SIAM J. Comput.* **34** (2005) 803–824.

[4] M. Charikar, S. Guha, E. Tardos and D. Shmoys, A constant-factor approximation algorithm for the k-median problem. *J. Comput. Syst. Sci.* **65** (2002) 129–149.

[5] G. Cornuejols, M.L. Fisher and G.L. Nemhauser, Exceptional paper-location of bank accounts to optimize float: An analytic study of exact and approximate algorithm. *Manage. Sci.* **23** (1977) 789–810.

[6] A. Gupta and K. Tangwongsan, Simpler Analyses of Local Search Algorithms for Facility Location. Preprint arXiv:0809.2554 (2008).

[7] K. Jain and V. Vazirani, Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *J. ACM* **48** (2001) 274–296.

[8] K. Jain, M. Mahdian, E. Markakis, A. Saberi and V. Vazirani, Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP. *J. ACM* **50** (2003) 795–824.

[9] A. Klose and A. Drexl, Facility location models for distribution system design. *Eur. J. Oper. Res.* **162** (2005) 429.

[10] S. Li, A 1.488 approximation algorithm for the uncapacitated facility location problem. *Inf. Comput.* **222** (2013) 45–58.

[11] L. Qiu, V. Padmanabhan and G. Voelker, On the Placement of Web Server Replicas, in *Proc. of IEEE INFOCOM01* (2001) 1587–1596.

[12] D. Shmoys, E. Tardos and K. Aardal, Approximation Algorithms for Facility Location Problems, in *Proc. of the 29th Annual ACM Symposium on Theory of Computing* ACM. New York (1997) 265–274.

[13] P. Zhang, A new approximation algorithm for the k-facility location problem. *Theoret. Comput. Sci.* **384** (2007) 126–135.

Communicated by J. Hromkovic.
Received October 8, 2014. Accepted February 25, 2016.