AN ADAPTIVE PROBABILISTIC ALGORITHM FOR ONLINE $k$-CENTER CLUSTERING

RUIQI YANG, DACHUAN XU AND YICHENG XU

Department of Information and Operations Research, College of Applied Sciences
Beijing University of Technology, Beijing 100124, China

DONGMEI ZHANG*
School of Computer Science and Technology
Shandong Jianzhu University, Jinan 250101, China

(Communicated by Wenxun Xing)

Abstract. The $k$-center clustering is one of the well-studied clustering problems in computer science. We are given a set of data points $P \subseteq \mathbb{R}^d$, where $\mathbb{R}^d$ is $d$ dimensional Euclidean space. We need to select $k \leq |P|$ points as centers and partition the set $P$ into $k$ clusters with each point connecting to its nearest center. The goal is to minimize the maximum radius. We consider the so-called online $k$-center clustering model where the data points in $\mathbb{R}^d$ arrive over time. We present the bi-criteria $(\frac{n}{k}, (\log U^* \log L^*)^2)$-competitive algorithm and $(\frac{n}{k}, \log \gamma \log \frac{n}{k})$-competitive algorithm for semi-online and fully-online $k$-center clustering respectively, where $U^*$ is the maximum cluster radius of optimal solution, $L^*$ is the minimum distance of two distinct points of $P$, $\gamma$ is the ratio of the maximum distance of two distinct points and the minimum distance of two distinct points of $P$ and $n$ is the number of points that will arrive in total.

1. Introduction. In the $k$-center clustering problem, we are given a set of $n$ points $P \subseteq \mathbb{R}^d$ and an integer $k$. We need to choose a center set $C$ containing $k$ points in $P$ which are called centers, and minimize the distance from all points in $P$ to their nearest center in $C$, i.e.,

$$\min_{C \subseteq P, |C| \leq k} \max_{p \in P} \min_{c \in C} \|c - p\|.$$ 

The $k$-center clustering is proved $NP$-complete by Gary and Johnson [17]. The lower bound of approximation factor is 2 unless $P = NP$ (cf. [11, 8, 10, 12]). Bar-Ilan et al. [2] introduced the capacitated $k$-center clustering where each center has a capacity, and proposed a 10-approximation algorithm for the uninform capacitated version. Khuller and Sussmann [14] then improved it to 6-approximation and gave a 5-approximation algorithm for the soft uniform version that multiple centers may be chosen at the same location. Cygan et al. [6] proposed the first constant approximation algorithm for the non-uniform capacitated $k$-center clustering. Cygan et al. proved that there are no approximation algorithms with approximation factor less than 3 unless $P = NP$.

2010 Mathematics Subject Classification. Primary: 90C27; Secondary: 68W27.
Key words and phrases. $k$-center, online clustering, probabilistic algorithm.

* Corresponding author: Dongmei Zhang.
Krumke [15] introduced the fault-tolerant $k$-center clustering where each point must be assigned to a given number of centers. In this model, any subset of centers with size $\alpha$ may fail to serve clients. The goal is to minimize the maximum distance among each point to its $\alpha + 1$ nearest center. The results are presented [15, 4, 14]. In the capacitated fault-tolerant $k$-center clustering, the capacities of centers are limited and centers may fail to serve demand points. Chechik and Peleg [5] gave a 9-approximation and 17-approximation algorithm for the uniform capacitated and conservative fault-tolerant $k$-center clustering respectively. For the conservative capacitated fault-tolerant model, only those points that had been assigned to faulty centers may be reassigned. Under uniform version, Fernandes et al. [9] improved these results to 6-approximation and 7-approximation respectively. They also gave the first constant approximations for the non-uniform capacitated and conservative fault-tolerant $k$-center clustering.

**Our contributions and techniques.** Our results can be stated as follows.

1. Given a semi-online $k$-center clustering, we propose an $O(n^k)$-competitive algorithm producing a solution with the number of centers at most $O(k \log \frac{U^*}{L^*})^2$ in expectation.
2. Given a fully-online $k$-center clustering, we propose an $O(n^k)$-competitive algorithm producing a solution with the number of centers at most $O(k \log \gamma \log \frac{\gamma}{k})$ in expectation.

We are motivated by applications of documents and images classification for information retrieval. Given $d$, let us consider a sequence of points from $R^d$ in which each point comes one by one. At the beginning, neither the total number of points nor the order information of coming points are known. But when a point comes, we need to decide immediately if assign it to one of the current clusters or initiate the point as a new center. However, there is a $k$ restriction on the total number of centers. The goal is to minimize the maximum distance of the arrival points to their nearest center. The above model is quite relevant to the incremental clustering problem proposed by Charikar [3]. They obtained deterministic as well as randomized algorithms for the incremental clustering and proved the inapproximation bounds for deterministic incremental clustering algorithm and randomized incremental algorithm are 3 and $3 - \epsilon$ respectively for any fixed $\epsilon > 0$. Analogous with the incremental clustering, we consider the case that the number of points in $P \subseteq R^d$ is known. Inspired by Liberty et al. [16], we call it semi-online $k$-center clustering (SOKCC). We give a probabilistic algorithm for it. Through the technical analysis, we prove that the expected number of centers has a good upper bound. We show that the maximum distance of each point to its nearest center can also be bounded. If the number of points that will arrive is unknown, the problem turns to the so called fully online $k$-center clustering (FOKCC). By modifying the above algorithm, we show that there exists upper bounds for the number of centers and the maximum cluster diameter as well.

**Organizations.** We present necessary preliminaries in Section 2. In Section 3, we propose the adaptive probabilistic algorithm for the SOKCC. We prove that in this algorithm the number of expectation of centers and cluster diameter are bounded in Theorems 3.3 and 3.5. In Section 4, we propose an algorithm for the FOKCC and the corresponding bounds are given in Theorems 4.1 and 4.4. Concluding remarks are given in Section 5.
2. Preliminaries. We will give an example to show that we may suffer unbounded approximation ratio if we always maintain the cardinality of center for the online $k$-center clustering. All points are located in number axis, where the point is placed at location $x^{2t}$ at moment $t$. Let $x(\geq 2)$ be an integer and $k$ fixed to 2. If the point at $x^{2t}$ is clustered in existing centers by any online algorithm, the maximum radius is at least $(x^2 - 1) \cdot x^{2t - 1}$. For an off-line algorithm, the optimal clustering is that the point at $x^{2t}$ is clustered as one cluster and the previous points are clustered as another. Then the competitive ratio is at least $\Omega(x^2)$ (see Fig. 1).

To balance the number of the clusters and the competitive ratio of an online algorithm, we give a definition of bi-criteria competitive ratio for online algorithm (cf. Definition 2.2).

**Figure 1.** An example of a sequence of points $1, x^2, \ldots, x^{2t}$ coming over time. The circles with dotted lines are the clusters produced by an online algorithm. Solid ones are those produced by the optimal off-line algorithm.

**Definition 2.1.** ([1],[7]) Given a minimization clustering problem, a randomized online algorithm $A$ is $\rho$-competitive if the algorithm always returns a solution $C$ with

$$E\left[\max_{p \in P} \|p - C\|\right] \leq \rho \ast \text{opt}(I)$$

for any instance $I$, where $\text{opt}(I)$ denotes the optimal value for instance $I$.

**Definition 2.2.** Given a minimization clustering problem, a randomized online algorithm $A$ is bi-criteria $(\rho, \varrho)$-competitive if it always returns a solution $C$ with

$$E\left[\max_{p \in P} \|p - C\|\right] \leq \rho \ast \text{opt}(I) \quad \text{and} \quad E[|C|] \leq \varrho \ast k$$

for any instance $I$.

**Notations.** We introduce some notations as follows.

- $[k] := \{1, 2, \ldots, k\}$;
- Let $\{c_1, \ldots, c_k\}$ be any solution to $k$-center clustering;
- Let $\{c_1^*, \ldots, c_k^*\}$ be an optimal solution to $k$-center clustering;
- $C := \{c_1, \ldots, c_k\}$;
- $C_{OPT} := \{c_1^*, \ldots, c_k^*\}$;
\[ S_i := \{ p \in P : \| p - c_i \| \leq \| p - c_j \| \forall j \neq i, i, j \in [k] \}; \]

\[ S_i^* := \{ p \in P : \| p - c_i^* \| \leq \| p - c_{i'}^* \|, \forall c_{i'} \in C_{OPT} \}; \]

\[ U_i^* := \max_{p \in S_i^*} \| p - c_i^* \|; \]

\[ L_i^* := \min_{p \in S_i^* \setminus \{ c_i^* \}} \| p - c_i^* \|; \]

\[ U^* := \max_{i \in [k]} U_i^*; \]

\[ L^* := \min_{i \in [k]} L_i^*; \]

\[ \| p - C \| := \min_{c \in C} \| p - c \|. \]

Remark that \( S_i^* \) is the cluster corresponding to \( c_i^* \) and \( U_i^* \) (\( L_i^* \)) is an upper (lower) bound for cluster radius of \( S_i^* \). Though we do not know the optimal solution, we assume that the lower bound is \( L_i^* \) and the upper bound is \( U_i^* \). Then we can partition \( S_i^* \) for a given \( i \in [k] \) as \( S_{i,0}^* \) and \( S_{i,\tau}^* \), where

\[ S_{i,0}^* = \{ p \in S_i^* : \| p - c_i^* \| \leq L_i^* \}; \]

\[ S_{i,\tau}^* = \{ p \in S_i^* : 2^{-1} L_i^* < \| p - c_i^* \| \leq 2^\tau L_i^* \}, \text{ see Fig. 2.} \]

**Lemma 2.3.** Let \( S_{i,\tau(i)}^* \) be the part of partition with maximum index. For any \( i \in [k] \) and \( S_i^* \), \( \log \frac{U_i^*}{L_i^*} \leq \log \frac{U_i^*}{L_i^*} \) is an upper bound for \( \tau(i) \).

**Proof.** W.l.o.g, we suppose \( \log \frac{U_i^*}{L_i^*} \) is an integer (if not, we choose \( \lceil \log \frac{U_i^*}{L_i^*} \rceil \) be the bound). If

\[ \tau(i) > \log \frac{U_i^*}{L_i^*}, \]

and there exists a point \( p_0 \) in \( S_{i,\tau(i)}^* \). From the definition of \( S_{i,\tau(i)}^* \), we have

\[ \| p_0 - c_i^* \| > 2^{\tau(i)-1} L_i^* \geq 2 \log \frac{U_i^*}{L_i^*} L_i^* = U_i^*, \]

which contradicts the definition of \( U_i^* \). \( \Box \)

**Figure 2.** An illustration of partition of \( S_i^* \) for any given \( i \in [k] \)

3. **An adaptive probabilistic algorithm for the SO\( KCC.** Our algorithm is inspired by Liberty et al. [16]. During the algorithm, the probability of an arrival point selected as a center is rely on the distance between the point to a set that has been chosen as centers.

To control the number of centers in phase \( t \), we introduce an adaptive factor \( \alpha(t) \), where \( \alpha : N \to R^+ \) is a function of \( t \). For the definition of \( \alpha(t) \), we mean
that the adaptive factor $\alpha$ is update over the phase $t$. Thus, we let $\Pr\{v\} := \min\{(\alpha(t))^{-1} \cdot \|v - C\|, 1\}$ be the probability of point $v$ selected as a center. To control the amount of the centers produced in each phase, we present an integer $M$ as an upper bound. We also appoint to $\|v - \emptyset\| = +\infty$. Once we add a point $v$ to $C$ and the number of the centers is more than $M$ during phase $t$, we will update the iteration to phase $t+1$ and add multiple times of the adaptive factor $\alpha(t)$, i.e., $\alpha(t+1) = \beta \cdot \alpha(t)$, where $\beta \geq 2$ is a given integer. Repeat the above procedure until each arrival point is either selected as a center or assigned to one of the existing clusters. We present the detail as follows.

**Algorithm 1** An adaptive probabilistic algorithm for the SOKCC

**Input:** an integer $k$ and data number $n$

**Output:** centers set $C$

```
1: Initialize $t \leftarrow 1$, $C^t \leftarrow \emptyset$, $\alpha(1) \leftarrow \frac{nU^*}{k}$, $C \leftarrow \emptyset$
2: while there is a arriving point $v$ do
3:    Update $C^t \leftarrow C^t \cup \{v\}$ with probability $\Pr\{v\} := \min\{(\alpha(t))^{-1} \cdot \|v - C\|, 1\}$
4:    $C \leftarrow C \cup C^t$
5:    if $|C^t| \geq 3k(1 + \log \frac{U^*}{L_i})$ then
6:        $t \leftarrow t + 1$
7:    end if
8: end while
9: return $C$
```

**Proposition 3.1.** For any $i \in [k]$, let $p \in S^*_{i,\tau}$ be the first point that is selected as a center during phase $t$. Then the probability of any other point $p' \in S^*_{i,\tau}$ arriving after point $p$ selected as a center satisfies

$$\Pr\{p'\} \leq (\alpha(t))^{-1} \cdot 2^{t+1} L_i^*.$$

**Proof.** Let $C$ be the center set before point $p'$ arrives. By the choice of $p \in C$, the triangle inequality, and the definition of $S^*_{i,\tau}$, we have

$$\Pr\{p'\} := \min\{(\alpha(t))^{-1} \cdot \|p' - C\|, 1\} \leq (\alpha(t))^{-1} \cdot \|p' - p\| \leq (\alpha(t))^{-1} \cdot \|p' - c_i^*\| + (\alpha(t))^{-1} \cdot \|c_i^* - p\| \leq (\alpha(t))^{-1} \cdot 2^{t+1} L_i^*.$$

**Lemma 3.2.** Let $t'$ be the first phase satisfying $\alpha(t) \geq nU^*/k$. If Algorithm 1 terminates after $t'$, we have

$$E[|C \cap S^*_i|] \leq 1 + \log \frac{U_i^*}{L_i} + 12(\alpha(t'))^{-1} U_i^* |S^*_i|, \quad \forall i \in [k].$$

**Proof.** For any $i \in [k]$, let $p \in S^*_{i,\tau}$ be the first point that is selected as a center and $S^*_{i,\tau,t}$ be set of points in $S^*_{i,\tau}$ that the algorithm encounters during phase $t$. From Proposition 3.1, the expectation of the number of centers chosen from $S^*_{i,\tau}$ during and after $t'$ is bounded by

$$1 + \sum_{t \geq t'} (\alpha(t))^{-1} \cdot 2^{t+1} L_i^* |S^*_{i,\tau,t}|.$$
Thus, we have

\[
E[|C \cap S^*_i|] \leq \sum_{\tau:=0} \left( 1 + \sum_{t \geq t'} (\alpha(t))^{-1} \cdot 2^{\tau+1} L^*_i |S^*_i,\tau,t| \right).
\]

For any \(S^*_i, i \in [k]\), we have

\[
\sum_{\tau:=0} \left( 1 + \sum_{t \geq t'} (\alpha(t))^{-1} \cdot 2^{\tau+1} L^*_i |S^*_i,\tau,t| \right)
\leq 1 + \log \frac{U^*_i}{L^*_i} + \sum_{\tau:=0} \sum_{t \geq t'} (\alpha(t))^{-1} \cdot 2^{\tau+1} L^*_i |S^*_i,\tau,t|
\leq 1 + \log \frac{U^*_i}{L^*_i} + \sum_{\tau:=0} (\alpha(t'))^{-1} \cdot 2^{\tau+2} L^*_i |S^*_i,\tau|
\leq 1 + \log \frac{U^*_i}{L^*_i} + 12 (\alpha(t'))^{-1} L^*_i |S^*_i,0| + 8 (\alpha(t'))^{-1} \sum_{\tau:=1} 2^{\tau(i)-1} L^*_i |S^*_i,\tau|
\leq 1 + \log \frac{U^*_i}{L^*_i} + 12 (\alpha(t'))^{-1} U^*_i |S^*_i|.
\]

\[
\text{Theorem 3.3.} \quad \text{Let } C \text{ be the center set produced by Algorithm 1. We have}
\]

\[
E[|C|] = O \left( k \left( \frac{U^*}{L^*} \right)^2 \right).
\]

**Proof.** We proceed the proof in the following two possibilities.

1. Algorithm 1 terminates before phase \(t'\), where \(t'\) is the first iteration \(t\) such that \(\alpha(t) \geq \frac{\log \frac{U^*}{L^*}}{k}\). We know that the algorithm produces at most \(3k(1 + \log \frac{U^*}{L^*})\) centers in each iteration and the adaptive factor from \(\alpha(1)\) to \(\alpha(t'-1)\) needs at most \(\frac{t'}{t'}\) iterations. The number of the centers produced before \(t'\) is bounded by \(O(k(\log \frac{U^*}{L^*})^2)\).

2. Algorithm 1 terminates at phase \(t'\) or \(t > t'\).

From Lemma 3.2, we sum the inequality (1) over \(i \in [k]\),

\[
\sum_{i \in [k]} E[|C \cap S^*_i|] \leq k \left( 1 + \log \frac{U^*}{L^*} \right) + 12 (\alpha(t'))^{-1} nU^*
\leq k \left( 1 + \log \frac{U^*}{L^*} \right) + 12k
\leq O \left( k \log \frac{U^*}{L^*} \right).
\]

The second inequality is obtained from the choice of phase \(t'\). 

\[
\square
\]
Lemma 3.4. ([16]) Given a sequence $X_1,...,X_n$ of $n$ independent experiments, each experiment succeeds with probability $p_i \geq \min \left\{ \frac{A_i}{B} \right\}$, where $B \geq 0$ and $A_i \geq 0$ for any $i \in [n]$. Let $t$ be the number of consecutive unsuccessful experiments before the first successful one. We have

$$E \left[ \sum_{i=1}^{t} A_i \right] \leq B.$$

Theorem 3.5. Let $U$ be the maximum cluster radius in the solution produced by Algorithm 1. We have

$$E[U] = O \left( \frac{nU^*}{k} \right).$$

Proof. W.o.l.g., we assume that $\{p_1,...,p_s\}$ is the arriving sequence of points in cluster $S^*_i$. Let $l$ be the random number of consecutive points of $S^*_i,\tau$ before the first point is selected as a center. And assume $T$ is the last phase. Then any point $p_i \in S^*_i,\tau$ has probability

$$\Pr\{p_i\} = \min \{ (\alpha(t))^{-1} \cdot \|p_i - C\|, 1 \} \geq (\alpha(T))^{-1} \cdot \|p_i - C\|,$$

where $C$ is the center set before $p_i$ arrives. By Lemma 3.4,

$$E \left[ \max_{p_j \in [t]} \|p_j - C\| \right] \leq E \left[ \sum_{j=1}^{t} \|p_j - C\| \right] \leq \alpha(T).$$

Traversing all $\tau(i)$ and $i \in [k]$, we have

$$\max_{i \in [k]} \max_{\tau(i) \geq 0} E \left[ \max_{p_j, j \in [t]} \|p_j - C\| \right] = \max_{i \in [k]} \max_{\tau(i) \geq 0} \max_{j \in [t]} \|p_j - C\| \leq \alpha(T).$$

Let $t''$ be the first phase $t$ such that

$$\alpha(t) \geq \frac{36nU^*}{k}.$$ 

From the inequality (1), the number of centers during and after $t''$ is bounded by

$$1 + \log \frac{U^*}{L^*} + 12 (\alpha(t''))^{-1} nU^* \leq \frac{4k}{3} \left( 1 + \log \frac{U^*}{L^*} \right).$$

It follows from Algorithm 1 that proceeding the next phase of $t''$ is equivalent to producing more than $3k(1 + \log \frac{U^*}{L^*})$ centers during phase $t''$. By the Markov's inequality, the probability of Algorithm 1 proceeding the next phase of $t''$ is at most

$$\frac{4k}{3k} \left( 1 + \log \frac{U^*}{L^*} \right) \leq \frac{4}{9}.$$
The above inequality indicates that our algorithm terminates at phase $t''$ with probability at least $\frac{a}{9}$. Let $a$, $b$, $c$ be the probability of Algorithm 1 terminating before, at and after phase $t''$ respectively. We have

$$\Pr\{T = t\} := \begin{cases} 
a & \text{if } t < t'', \\
b & \text{if } t = t'', \\
c & \text{if } t > t''.
\end{cases} \quad (2)$$

Then we have

$$E[\alpha(T)] \leq a \cdot \alpha(t'' - 1) + (1 - a)E[\alpha(T)|t \geq t''] \leq \alpha(t'') + \alpha(t'')\Pr\{T = t''\} + \alpha(t'' + 1)\Pr\{T = t'' + 1\} + \ldots + \alpha(t)\Pr\{T = t\} \leq 2\alpha(t'') + \alpha(t'')\sum_{i=1}^{\frac{8}{9}} \left(\frac{8}{9}\right)^i \leq O(\alpha(t'')) \quad (3)$$

where $E[\alpha(T)|t \geq t'']$ denotes the expectation of $\alpha(T)$ after phase $t''$. The third inequality is obtained by the updating rule of $\alpha(t)$.

If $p_m$ is the first point of $S^*_{t,\tau}$ that is selected as a center by Algorithm 1, then the distance between any point $p_n \in S^*_{t,\tau}$ that arriving after point $p_m$ to its nearest center is bounded by

$$\|p_n - p_m\| \leq \|p_n - c^*_i\| + \|c^*_i - p_m\| \leq 2^{\tau + 1}L^*_i.$$

Enumerating all $i \in [k]$ and $\tau \in [\lceil \log \frac{L^*_i}{L^*_1} \rceil]$, we have

$$\max_{i \in [k]} \max_{\tau \geq 0} \min_{p_n \in S^*_{t,\tau}} \|p_n - C\| \leq \max_{i \in [k]} \max_{\tau \geq 0} 2^{\tau + 1}L^*_i \leq O(U^*).$$

4. The FOKCC. In this section, we consider the model where the number information of coming points is unknown. We modify the algorithm in the section 3 to deal with the FOKCC.

Analogously to Liberty [16], we choose the first arrival $k + 1$ distinct points as centers. Note we can only utilize the current points information to obtain $\alpha(t)$.

Notations. We introduce some notations to simplify the analysis.

- Let $n$ denote a number counter of the arrived points from $P$;
- Let $n_i$ denote the number of points arriving at the beginning of phase $t$;
- Let $L^{(t)} := \min_{p, p' \in P(t)} \|p - p'\|$ be the minimum distance among $P(t)$, where $P(t)$ is the point set at the end of phase $t$;
- Let $U^{(t)} := \max_{p, p' \in S^{(t)}_i} \|p - p'\|$ be the maximum distance among $S^{(t)}_i$, where $S^{(t)}_i$ is the $i$th optimal cluster point set corresponding to $P(t)$;
- Let $U^{(t)} := \max_{i \in [k]} U^{(t)}$ be the maximum cluster radius of an optimal solution corresponding to phase $t$;
- Let $q_t$ denote the number centers produced by the algorithm during phase $t$.

Initially, we set

$$\alpha(1) := \frac{n_1 L^{(1)}}{k},$$
where \( n_1 = k + 1 \). For an arrival point \( v \),
\[
\Pr\{v\} := \min \left\{ (\alpha(t))^{-1} \cdot \|v - C\|, 1 \right\}.
\]

We introduce \( M(t) \) as an upper bound for the centers produced during phase \( t \). Let
\[
M(t) := 3k \left( 1 + \log \frac{U^{(t)*}}{L^{(t)*}} \right).
\]

Once \( q_t \geq M(t) \) at iteration \( t \), update \( t := t + 1 \) and \( \alpha(t + 1) := \beta \alpha(t) \). Repeat the above process until all points have been assigned. See Algorithm 2.

**Algorithm 2** An adaptive probabilistic algorithm for the FOKCC

**Input:** an integer \( k \)

**Output:** center set \( C \)

1. To initialize, choose the first \( k + 1 \) distinct data points as center set \( C \). Let \( n_1 := k + 1, \ C^t := \emptyset \) be the centers set at the beginning of phase \( t \), \( P^{(t)} := \emptyset \), \( \alpha(1) := \frac{n_1 L^{(1)\ast}}{k} \).
2. while there is a data point \( v \) do
3. Update \( n := n+1 \)
4. Update \( P^{(t)} := P^{(t)} \cup \{v\} \)
5. Update \( C^t := C^t \cup \{v\} \) with probability \( \Pr\{v\} := \min\{ (\alpha(t))^{-1} \cdot \|v - C\|, 1 \} \)
6. Update \( C := C \cup C^t \)
7. if \( q_t \geq 3k(1 + \log \frac{U^{(t)*}}{L^{(t)*}}) \) then
8. Update \( t := t + 1 \)
9. Update \( \alpha(t + 1) := 2 \cdot \alpha(t) \)
10. end if
11. end while
12. return \( C \)

We give an upper bound for the expectation number of centers as follows.

**Theorem 4.1.** Let \( C \) be the centers produced by Algorithm 2. We have
\[
E[|C|] \leq O \left( k \log \frac{U^*}{L^*} \log \frac{nU^*}{(k+1)L^*} \right),
\]
where \( n \) is the number of point set \( P^{(t)} \) and \( U^*(L^*) \) is the maximum (minimum) distance among the point set \( P^{(t)} \) in the end.

**Proof.** Let \( t' \) be the first phase \( t \) such that
\[
\alpha(t) \geq \frac{n_t U^{(t)*}}{k}.
\]

If Algorithm 2 terminates before \( t' \), we conclude the proof analogous to Theorem 3.3. Now we consider the case that Algorithm 2 terminates at or after phase \( t' \). From the choice of \( \alpha(t') \) and the proof of inequality (1), the number of the centers produced during and after iteration \( t' \) is at most
\[
k \left( 1 + \log \frac{U^{(t')\ast}}{L^{(t')\ast}} \right) + \frac{12n_t U^{(t')\ast}}{\alpha(t')} \leq O \left( k \log \frac{U^*}{L^*} \right).
\]

\( \square \)
Corollary 4.2. Let 
\[ \gamma := \max_{v, v' \in P_t} \|v - v'\| \min_{v, v' \in P_t} \|v - v'\|, \]
where \( P_t \) is the arrival point set in the end. We have 
\[ E[|C|] \leq O\left(k \log \gamma \log \frac{\gamma n}{k}\right). \]

Lemma 4.3. Let \( \{S_{i, \tau}(t)\}_{\tau=0} \) be the partition of the cluster \( S_i^{(t)*} \), where \( S_i^{(t)*} \) is the \( i \)th optimal cluster at the end of phase \( t \). Setting \( q'_t \) to be the number of points selected as centers in \( S_i^{(t)*} \) but are not first chosen as the centers during phase \( t \), we have 
\[ q_t \leq k \left(1 + \log \frac{U^{(t)*}}{L^{(t)*}}\right) + q'_t. \]

Proof. From Lemma 2.3, we have the upper bound \( \log \frac{U^{(t)*}}{L^{(t)*}} + 1 \) for \( \tau \). If each part of the partition of \( S_i^{(t)*} \) has a point selected as a center, we have 
\[ q'_t \geq q_t - k \left(1 + \log \frac{U^{(t)*}}{L^{(t)*}}\right). \]

There is an upper bound for the maximum distance among all the arrival points to their nearest centers in expectation.

Theorem 4.4. Let \( U \) be the maximum radius of clusters among each arrival point to its nearest center in Algorithm 2. We have 
\[ E[U] = O\left(\frac{nU^*}{k}\right). \]

Proof. Consider \( p_0 \) as the first point of \( S_i^{(t)*} \) selected as center during phase \( t \) in Algorithm 2. For any other arriving point \( p' \) in \( S_i^{(t)*} \), we have 
\[ \max_{p' \in S_i^{(t)*}} \|p' - C\| \leq \|p' - p_0\| \leq 2^{\tau+1}L_i^{(t)*}, \]
where \( C \) is the center set before \( p' \) arrives. Traversing all \( \tau \) and \( i \in [k] \), the maximum cluster radius is bounded by \( O(U^*) \).

W.o.l.g., we assume 
\[ \{S_i^{(t)*}\} = \{p_1, p_2, ..., p_m\}. \]

Let \( l(\leq m) \) be the number of continuous points of \( S_i^{(t)*} \) that are not chosen as centers before the first point is selected as a center. Based on Theorem 3.3 and Lemma 3.2, we only need to estimate the expectation value of \( \alpha(T) \).

Let \( t'' \) be the first phase \( t \) such that 
\[ \alpha(t) \geq \frac{36mU^{(t)*}}{k}. \]
From Lemma 4.3 and the choice of \( t'' \), we have 
\[ E[q_{t''}] \leq 12knU^{(t)*} \alpha(t'')^{-1} \leq \frac{k}{3}. \]
For Algorithm 2, the condition that one can process the next phase of $t''$ is
\[ q'_{t''} \geq 2k \left( 1 + \log \frac{U_{t''}}{L_{t''}} \right). \]
By the Markov’s inequality, the probability of Algorithm 2 processing the next phase of $t''$ is at most
\[ \frac{k}{2 \left( 1 + \log \frac{U_{t''}}{L_{t''}} \right)} \leq \frac{1}{6}. \]
Similar to Theorem 3.5, we give the estimate of the expectation of $\alpha(T)$,
\[ O(\alpha(t'')) = O \left( \frac{nU^*}{k} \right). \]

5. **Concluding remarks.** In this paper, we consider a so-called online $k$-center clustering model. To balance the quality of solution and the number of centers, we present two probability algorithms. The expectation of the number of centers is bounded by $O(k(\log U^*)^2)$ and the expectation of the maximum cluster radius is at most $O(\frac{nU^*}{k})$ for the semi-online version in Algorithm 1. And the corresponding bounds are $O(k \log \gamma \log \frac{nU}{k})$ and $O(\frac{nU^*}{k})$ respectively for the fully-online version in Algorithm 2. Furthermore, we show that we are not able to bound the competitive ratio if we are required to satisfy $k$-restriction strictly for both semi-online and fully-online $k$-center clustering (an example see Fig. 1). It remains an open problem if there exists a constant bi-criteria competitive ratio for the above two online $k$-center clustering problems.

**Acknowledgments.** The research of the first three authors are supported by Natural Science Foundation of China (No. 11531014). The fourth author’s research is supported by Science and Technology Development Plan Project of Jinan City (No. 201401211) and Higher Educational Science and Technology Program of Shandong Province (No. J15LN23).

**REFERENCES**

[1] D. Achlioptas, M. Chrobak and J. Noga, Competitive analysis of randomized paging algorithms, *Theoretical Computer Science*, 234 (2000), 203–218.

[2] J. Bar-Ilan, G. Kortsarz and D. Peleg, How to allocate network centers, *Journal of Algorithms*, 15 (1993), 385–415.

[3] M. Charikar, C. Chekuri, T. Feder and R. Motwani, Incremental clustering and dynamic information retrieval, *SIAM Journal on Computing*, 33 (2004), 1417–1440.

[4] S. Chaudhuri, N. Garg and R. Ravi, The $p$-neighbor $k$-center problem, *Information Processing Letters*, 65 (1998), 131–134.

[5] S. Chechik and D. Peleg, The fault-tolerant capacitated $k$-center problem, *Theoretical Computer Science*, 566 (2015), 12–25.

[6] M. Cygan, M. T. Hajiaghayi and S. Khuller, LP rounding for $k$-centers with non-uniform hard capacities, *Proceedings of the fifty-third Annual Symposium on Foundations of Computer Science (FOCS)*, IEEE, (2012), 273–282.

[7] S. B. David, A. Borodin, R. Karp, G. Tardos and A. Wigderson, On the power of randomization in on-line algorithms, *Algorithmica*, 11 (1994), 2–14.

[8] T. Feder and D. Greene, Optimal algorithms for approximate clustering, *Proceedings of the twentieth Annual ACM Symposium on Theory of Computing*, ACM, (1988), 434–444.

[9] C. G. Fernandes, S. P. de Paula and L. L. C. Pedrosa, Improved approximation algorithms for capacitated fault-tolerant $k$-center, *Algorithmica*, 80 (2018), 1041–1072.
[10] T. F. Gonzalez, Clustering to minimize the maximum intercluster distance, *Theoretical Computer Science*, 38 (1985), 293–306.

[11] D. S. Hochbaum and D. B. Shmoys, A best possible heuristic for the $k$-center problem, *Mathematics of Operations Research*, 10 (1985), 180–184.

[12] W. L. Hsu and G. L. Nemhauser, Easy and hard bottleneck location problems, *Discrete Applied Mathematics*, 1 (1979), 209–215.

[13] S. Khuller, R. Pless and Y. J. Sussmann, Fault tolerant $k$-center problems, *Theoretical Computer Science*, 242 (2000), 237–245.

[14] S. Khuller and Y. J. Sussmann, The capacitated $k$-center problem, *SIAM Journal on Discrete Mathematics*, 13 (2000), 403–418.

[15] S. O. Krumke, On a generalization of the $p$-center problem, *Information Processing Letters*, 56 (1995), 67–71.

[16] E. Liberty, R. Sriharsha and M. Sviridenko, An algorithm for online $k$-means clustering, *Proceedings of the Eighteenth Workshop on Algorithm Engineering and Experiments (ALENEX)*, Society for Industrial and Applied Mathematics, (2016), 81–89.

[17] R. G. Michael and S. J. David, Computers and intractability: A guide to the theory of NP-completeness, *W. H. Freeman, San Francisco*, (1979), 90–91.

Received August 2017; revised December 2017.

E-mail address: yangruiqi@emails bjut.edu.cn
E-mail address: xudc@bjut.edu.cn
E-mail address: ycxu@emails bjut.edu.cn
E-mail address: zhangdongmei@sdjzu.edu.cn