A Extension to k-Means and (k, p)-Clustering

As stated in [20, 8], while [27] only discusses the k-median problem, their construction can easily be modified to work for k-means clustering and further generalized to work for (k, p)-clustering, where the (k, p)-clustering problem is defined in the same way as k-median problem except that we want to minimize $\sum_{x \in U} d(x, S)^p$ for some $S \subseteq U$ of size at most k. Note that $(k, 1)$-clustering and $(k, 2)$-clustering correspond to k-median and k-means respectively.

We define a ρ-metric space (U, d) in the same way as a metric space except for relaxing the condition that d must satisfy the triangle inequality to the condition that $d(x, y) \leq \rho(d(x, z) + d(z, y))$ for all $x, y, x \in U$. Given a metric space (U, d) and some $p \geq 1$, the results in Section 6 of [9] can easily be used to show that (U, d^p) is a 2^{p-1}-metric space, where $d^p(x, y)$ is defined to be $d(x, y)^p$.

We now show that the assignment σ maintained by our algorithm is $O(\rho^3)$-approximate when U is a ρ-metric space (i.e. that $\texttt{cost}(\sigma) = O(\rho^3) \cdot \texttt{opt}(U)$) and that the extraction technique of [18] can be generalized to ρ-metric spaces.

Lemma A.1. *When the underlying space U is a ρ-metric space, the assignment σ maintained by our algorithm is $O(\rho^3)$-approximate.*

Proof. By making the appropriate changes to the proofs of Lemma B.3 and Lemma B.4, we get generalizations of these lemmas to ρ-metric spaces, where the lemma statements are the same except for an extra ρ factor in the inequalities.

Lemma A.2. *Given any positive ξ, there exists a sufficiently large choice of α such that $\nu_i \leq 2\rho \cdot \mu_\gamma(U_i^{\text{OLD}})$ for each $i \in [t-1]$ with probability at least $1 - e^{-\xi k'}$.*

Lemma A.3. *Given metric subspaces U_1 and U_2 of U such that $|U_1 \oplus U_2| \leq \epsilon\gamma|U_1|$, we have that $\mu_\gamma(U_1) \leq 2\rho \cdot \mu_{\gamma^*}(U_2)$.*

These two lemmas immediately imply the following generalization of Lemma B.5.

Lemma A.4. *$\nu_i \leq 4\rho^2 \cdot \mu_{\gamma^*}(U_i)$ for each $i \in [t-1]$ with probability at least $1 - e^{-\xi k'}$.*

The upper bound on $\texttt{cost}(\sigma)$ given in Lemma B.6 can be generalized by noticing that $\texttt{cost}(\sigma, C_i) \leq 2\rho\nu_i|C_i|$ for all $i \in [t-1]$, which us that

$$\texttt{cost}(\sigma) \leq \sum_{i=1}^{t} 2\rho\nu_i|C_i|.$$

The lower bound on $\texttt{opt}(U)$ given in Lemma B.10 holds for ρ-metric spaces with no modifications. Hence, we get that with probability at least $1 - e^{-\xi k'}$ we have that

$$\texttt{cost}(\sigma) \leq \sum_{i=1}^{t} 2\rho\nu_i|C_i| \leq \sum_{i=1}^{t} 8\rho^3\mu_i|C_i| \leq \frac{16\rho^3 r}{1-\gamma^*}\,\texttt{cost}(S)$$

□

By making the appropriate modifications to the proof of Theorem C.1, we can extend this theorem to work for ρ-metric spaces. In particular, we can obtain a proof of Theorem A.5 by taking the proof of Theorem C.1 and adding extra ρ factors whenever the triangle inequality is applied.

Theorem A.5. *Given a ϕ-approximate m-assignment $\pi : U \to U$, any ψ-approximate solution to the weighted k-median instance $(\pi(U), d, w)$, where each point $x \in \pi(U)$ receives weight $w(x) := |\pi^{-1}(x)|$, is also a $\rho\left(\phi + 2(1+\phi)\psi\rho^2\right)$-approximate solution to the k-median instance (U, d) where U is a ρ-metric space.*

Since the algorithm in [26] is $O(1)$-approximate on $O(1)$-metric spaces, it immediately follows by applying Theorem A.5 and Lemma A.1 that our algorithm maintains a $O(1)$-approximate solution to the k-median problem on (U, d^p) for $p = O(1)$. Since the k-median problem on (U, d^p) is exactly the (k, p)-clustering problem on (U, d), it follows that our algorithm generalizes to solve instances of (k, p)-clustering in metric spaces.

B Proofs of Lemma 3.2 and Lemma 3.3

Throughout this section, we fix γ to be any real such that $\beta < \gamma < 1$ and ϵ to be any real such that $0 < \epsilon < \min\{\frac{1-\gamma}{2\gamma}, 1\}$. Let β^* and γ^* denote $\beta(1-\epsilon)$ and $\gamma(1+2\epsilon)$ respectively.

B.1 Proof of Lemma 3.2

We first prove Lemma B.1, which shows that the sizes of the sets U_i decrease exponentially with i.

Lemma B.1. *For all $i \in [t-1]$, $|U_{i+1}| \leq (1-\beta^*)|U_i|$.*

Proof. Consider the ratio $|U_{i+1}|/|U_i|$. Since $U_{i+1} \subseteq U_i$ and U_{i+1} is reconstructed every time U_i is reconstructed, it follows that $|U_{i+1}|/|U_i|$ is at most $(n_{i+1}+\ell)/(n_i+\ell-\ell')$, where n_j is the size of U_j at the time it was last reconstructed and ℓ and ℓ' are the number of insertions and deletions that have occurred since the last time U_{i+1} was reconstructed respectively. By Lemma B.2, we get that this expression is upper bounded by $(n_{i+1}+\tau n_{i+1})/n_i$. Now we can observe that

$$\frac{|U_{i+1}|}{|U_i|} \leq \frac{n_{i+1}+\ell}{n_i+\ell-\ell'} \leq \frac{n_{i+1}+\tau n_{i+1}}{n_i} \leq \frac{n_{i+1}}{n_i} + \tau \leq (1-\beta) + \epsilon\beta = 1-\beta^*,$$

where we use the facts that $n_{i+1} \leq (1-\beta)n_i$ and $\tau \leq \epsilon\beta$ in the final inequality.

Lemma B.2. *Given some integer $i \in [t-1]$, let ℓ and ℓ' be the number of insertions and deletions that have occurred since the last time U_{i+1} was reconstructed respectively. Then we have that*

$$\frac{n_{i+1}+\ell}{n_i+\ell-\ell'} \leq \frac{n_{i+1}+\tau n_{i+1}}{n_i}.$$

Proof. First, note that $(n_{i+1}+\ell)/(n_i+\ell-\ell') \leq (n_{i+1}+\ell)/(n_i-\ell')$. Now, given some reals $A \geq a \geq 0$ and $0 \leq N \leq A-a$, we define a function $f : [0,1] \to \mathbb{R}$ by $f(x) = (a+xN)/(A-(1-x)N)$. The derivative of f is $-N(a-A+N)/((x-1)N+A)^2$ and is non-negative for all $x \in [0,1]$. Hence, $f(x) \leq f(1)$ for all $x \in [0,1]$.

By setting $A = n_i$, $a = n_{i+1}$, $N = \ell + \ell'$ and noting that $\ell + \ell' \leq \tau n_{i+1}$ by Invariant 3.1 and $n_{i+1} \leq (1-\beta)n_i$, we get that

$$\ell + \ell' \leq \tau n_{i+1} \leq \beta n_{i+1} = (1+\beta)n_{i+1} - n_{i+1} \leq (1+\beta)(1-\beta)n_i - n_{i+1} \leq n_i - n_{i+1},$$

and hence it follows that

$$\frac{n_{i+1}+\ell}{n_i-\ell'} = f\left(\frac{\ell}{\ell+\ell'}\right) \leq f(1) = \frac{n_{i+1}+\ell+\ell'}{n_i} \leq \frac{n_{i+1}+\tau n_{i+1}}{n_i}.$$

□

□

Since $|U_1| = |U|$, $|U_{t-1}| > (1-\tau)\alpha k' = \Omega(k)$, and β^* is a constant, we infer that $t = O\left(\log \frac{|U|}{k}\right) = \tilde{O}(1)$.

B.2 Proof of Lemma 3.3

Bounding the Radii ν_i (Lemma B.5): Let U_i^{OLD} denote the state of the ith layer the last time it was reconstructed for $i \in [t]$. We now use the following crucial lemma which is analogous to Lemma 4.3.3 in [25].

Lemma B.3. *Given any positive ξ, there exists a sufficiently large choice of α such that $\nu_i \leq 2\mu_\gamma(U_i^{\mathrm{OLD}})$ for each $i \in [t-1]$ with probability at least $1 - e^{-\xi k'}$.*

Henceforth, we fix some positive ξ and sufficiently large α such that Lemma B.3 holds.

Lemma B.4. *Given metric subspaces U_1 and U_2 of U such that $|U_1 \oplus U_2| \leq \epsilon\gamma|U_1|$, we have that $\mu_\gamma(U_1) \leq 2\mu_{\gamma^*}(U_2)$.*[4]

Proof. Let X be a subset of U_2 of size k such that $\nu_{\gamma(1+2\epsilon)}(X, U_2) = \mu_{\gamma(1+2\epsilon)}(U_2)$, $\rho = \mu_{\gamma(1+2\epsilon)}(U_2)$, and $A = B_{U_1}(X, \rho)$. Now note that

$$\begin{aligned}
|A| &= |B_{U_1 \cup U_2}(X, \rho) \setminus B_{U_2 \setminus U_1}(X, \rho)| \\
&\geq |B_{U_2}(X, \rho)| - |B_{U_2 \setminus U_1}(X, \rho)| \\
&\geq \gamma(1 + 2\epsilon)|U_2| - |U_2 \setminus U_1| \\
&\geq \gamma(1 + 2\epsilon)|U_2| - \epsilon\gamma|U_1| \\
&\geq \gamma(1 + 2\epsilon)\left(|U_1| - \epsilon\gamma|U_1|\right) - \epsilon\gamma|U_1| \\
&= \gamma|U_1| + \epsilon\gamma(1 - \gamma(1 + 2\epsilon))|U_1| \\
&\geq \gamma|U_1|.
\end{aligned}$$

Since there also exists a subset $Y \subseteq A$ of size k such that $A \subseteq B_{U_1}(Y, 2\rho)$, it follows that $\nu_\gamma(Y, U_1) \leq 2\rho$. Hence, $\mu_\gamma(U_1) \leq \nu_\gamma(Y, U_1) \leq 2\mu_{\gamma(1+2\epsilon)}(U_2)$. □

Lemma B.5. *$\nu_i \leq 4\mu_{\gamma^*}(U_i)$ for each $i \in [t-1]$ with probability at least $1 - e^{-\xi k'}$.*

Proof. For each $i \in [t-1]$, $|U_i \oplus U_i^{\text{OLD}}| \leq \tau|U_i^{\text{OLD}}|$ since, by Invariant 3.1, at most $\tau|U_i^{\text{OLD}}|$ points have been inserted or deleted from U_i since it was last reconstructed. Noticing that $\tau \leq \epsilon\gamma$, we can see that

$$|U_i \oplus U_i^{\text{OLD}}| \leq \epsilon\gamma|U_i^{\text{OLD}}|.$$

By now applying Lemma B.4 it follows that $\mu_\gamma(U_i^{\text{OLD}}) \leq 2\mu_{\gamma^*}(U_i)$. The lemma follows by combining this result with Lemma B.3. □

Upper Bounding $\texttt{cost}(\sigma)$ (Lemma B.6):

Lemma B.6.

$$\texttt{cost}(\sigma) \leq \sum_{i=1}^{t} 2\nu_i|C_i|.$$

Proof. We first note that for all $i \in [t-1]$, $\texttt{cost}(\sigma, C_i) \leq 2\nu_i|C_i|$. This follows directly from the fact that each point x in C_i is assigned to some point $y \in C_i$ such that $d(x, y) \leq 2\nu_i$. Since the C_i partition U and $\texttt{cost}(\sigma, C_t) = 0$, we get:

$$\texttt{cost}(\sigma) = \sum_{i=1}^{t} \texttt{cost}(\sigma, C_i) \leq \sum_{i=1}^{t} 2\nu_i|C_i|.$$

□

Lower Bounding $\texttt{opt}(U)$ (Lemma B.10): Let r denote $\lceil \log_{1-\beta^*} \frac{1-\gamma^*}{3} \rceil$ and for each $i \in [t]$ let μ_i denote $\mu_{\gamma^*}(U_i)$.

For the rest of this subsection we fix an arbitrary $S \subseteq U$ of size k. For each $i \in [t]$, let F_i denote the set $\{x \in U_i \mid d(x, S) \geq \mu_i\}$, and for any integer $m > 0$, let F_i^m denote $F_i \setminus (\cup_{j>0} F_{i+jm})$ and $G_{i,m}$ denote the set of all integers $j \in [t]$ and $j \equiv i \pmod{m}$.

Lemma B.7. *Given some $i \in [t]$ and a subset $X \subseteq F_i$, we have that $|F_i| \geq (1 - \gamma^*)|U_i|$ and $\texttt{cost}(S, X) \geq \mu_i|X|$.*

Proof. It follows directly from the definition of μ_i that we have that $|F_i| \geq (1 - \gamma^*)|U_i|$. By the definition of F_i, we have that $\texttt{cost}(S, X) = \sum_{x \in X} d(x, S) \geq \mu_i|X|$. □

The following lemma is proven in [25].

[4] $\oplus$ denotes symmetric difference, i.e. $U_1 \oplus U_2 = (U_1 \setminus U_2) \cup (U_2 \setminus U_1)$.

Lemma B.8 ([25], Lemma 4.3.8)**.** *Given integers $\ell \in [t]$ and $m > 0$, we have that*

$$\texttt{cost}(S, \cup_{i \in G_{\ell,m}} F_i^m) \geq \sum_{i \in G_{\ell,m}} \mu_i |F_i^m|.$$

Lemma B.9. *For all $i \in [t-1]$, we have that $|F_i^r| \geq \frac{1}{2}|F_i|$.*

Proof. We first note that for all $i \in [t-r]$, we have that $|F_{i+r}| \leq \frac{1}{3}|F_i|$. This follows from the fact that

$$|F_{i+r}| \leq |U_{i+r}| \leq (1-\beta^*)^r |U_i| \leq \frac{(1-\beta^*)^r}{1-\gamma^*}|F_i| \leq \frac{1}{3}|F_i|,$$

where the first inequality follows from the fact that $F_{i+r} \subseteq U_{i+r}$, the second inequality follows from Lemma B.1, the third inequality follows from Lemma B.7, and the fourth inequality follows from the definition of r. We now get that

$$|F_i^r| = |F_i \setminus \cup_{j>0} F_{i+jr}| \geq |F_i| - \sum_{j>0} \frac{1}{3^j}|F_i| \geq \frac{1}{2}|F_i|.$$

□

Lemma B.10.

$$\texttt{cost}(S) \geq \frac{1-\gamma^*}{2r} \sum_{i=1}^{t} \mu_i |C_i|.$$

Proof. Let $\ell = \arg\max_{0 \leq \ell < r} \{\sum_{i \in G_{\ell,r}} \mu_i |F_i^r|\}$. Then we have that

$$\texttt{cost}(S) \geq \texttt{cost}(S, \cup_{i \in G_{\ell,r}} F_i^r) \geq \sum_{i \in G_{\ell,r}} \mu_i |F_i^r| \geq \frac{1}{r} \sum_{i=1}^{t} \mu_i |F_i^r| \geq \frac{1}{2r} \sum_{i=1}^{t} \mu_i |F_i|$$
$$\geq \frac{1-\gamma^*}{2r} \sum_{i=1}^{t} \mu_i |U_i| \geq \frac{1-\gamma^*}{2r} \sum_{i=1}^{t} \mu_i |C_i|.$$

The second inequality follows from Lemma B.8, the third inequality from averaging and the choice of ℓ, the fourth inequality from Lemma B.9, and the fifth inequality from Lemma B.7. □

Proof of Lemma 3.3: It follows that with probability at least $1 - e^{-\xi k'}$ we have that

$$\texttt{cost}(\sigma) \leq \sum_{i=1}^{t} 2\nu_i |C_i| \leq \sum_{i=1}^{t} 8\mu_i |C_i| \leq \frac{16r}{1-\gamma^*} \texttt{cost}(S)$$

for any set $S \subseteq U$ of size k. Hence, we have that

$$\texttt{cost}(\sigma) \leq \frac{16r}{1-\gamma^*} \texttt{opt}(U).$$

C Proof of Corollary 3.4

In order to prove this corollary, we apply the extraction technique presented in [27] (with full details appearing in [25]) which is a slight generalization of the techniques from [18]. In particular, we use the following theorem which follows as an immediate corollary of Theorem 6 in [25]. For completeness, we provide a proof of this theorem.

Theorem C.1. *Given a ϕ-approximate m-assignment $\pi : U \to U$, any ψ-approximate solution to the weighted k-median instance $(\pi(U), d, w)$, where each point $x \in \pi(U)$ receives weight $w(x) := |\pi^{-1}(x)|$, is also a $(\phi + 2(1+\phi)\psi)$-approximate solution to the k-median instance (U, d).*

Proof. Let S^* be a solution to the weighted k-median instance $(\pi(U), d, w)$ and let S be an optimal solution to the k-median instance (U, d). Let ϕ and ψ be constants such that $\texttt{cost}(\pi, U) \leq \phi \cdot \texttt{opt}(U)$ and $\texttt{cost}_w(S^*, \pi(U)) \leq \psi \cdot \texttt{opt}_w(\pi(U))$. We now show that $\texttt{cost}(S^*, U) = O(1) \cdot \texttt{opt}(U)$. We first note that

$$\begin{aligned}\texttt{cost}(S^*, U) &= \sum_{x \in U} d(x, S^*) \\ &\leq \sum_{x \in U} d(x, \pi(x)) + \sum_{y \in \pi(U)} w(y) \cdot d(y, S^*) \\ &= \texttt{cost}(\pi, U) + \texttt{cost}_w(S^*, \pi(U)) \\ &\leq \phi \cdot \texttt{opt}(U) + \texttt{cost}_w(S^*, \pi(U)).\end{aligned}$$

Now note that, for any $X \subseteq U$ of size at most k, there exists some $Y \subseteq \pi(U)$ of size at most k such that $\texttt{cost}_w(Y, \pi(U)) \leq 2 \cdot \texttt{cost}_w(X, \pi(U))$. Since $\texttt{cost}_w(S^*, \pi(U)) \leq \psi \cdot \texttt{cost}_w(Y, \pi(U))$ for all $Y \subseteq \pi(Y)$ of size at most k, we get the following.

$$\begin{aligned}\texttt{cost}_w(S^*, \pi(U)) &\leq 2\psi \cdot \texttt{cost}_w(S, \pi(U)) \\ &= 2\psi \cdot \sum_{y \in \pi(U)} w(y) \cdot d(y, S) \\ &= 2\psi \cdot \sum_{x \in U} d(\pi(x), S) \\ &\leq 2\psi \cdot \sum_{x \in U} d(x, \pi(x)) + 2\psi \cdot \sum_{x \in U} d(x, S) \\ &= 2\psi \cdot \texttt{cost}(\pi, U) + 2\psi \cdot \texttt{opt}(U) \\ &\leq 2(1 + \phi)\psi \cdot \texttt{opt}(U).\end{aligned}$$

By combining these two chains of inequalities, we get that

$$\texttt{cost}(S^*, U) \leq \phi \cdot \texttt{opt}(U) + \texttt{cost}_w(S^*, \pi(U)) \leq (\phi + 2(1 + \phi)\psi) \cdot \texttt{opt}(U).$$

□

It immediately follows that we can get a $O(1)$-approximate solution to the instance (U, d) by running a static weighted k-median algorithm on the instance $(\sigma(U), d, w)$.

D Lower Bounds on Update and Query Time

In the static (i.e. non-dynamic) setting, the k-median problem is defined as follows: given a metric space U, return a set S of at most k points from U which minimizes the value of $\sum_{x \in S} d(x, S)$. The following lower bound for the static k-median problem is proven by Mettu in [25].

Theorem D.1. *Any $O(1)$-approximate randomized (static) algorithm for the k-median problem, which succeeds with even negligible probability, runs in time $\Omega(nk)$.*

Informally, the proof of this lower bound is obtained by constructing, for each $\delta > 0$, an input distribution of metric spaces (with polynomially bounded aspect ratio) on which no deterministic algorithm for the k-median problem succeeds with probability more than δ. Theorem D.1 then follows by an application of Yao's minmax principle.

We can use this lower bound from the static setting in order to get a lower bound for the dynamic setting. First note that any incremental algorithm for k-median with amortized update time $u(n, k)$ and query time $q(n, k)$ can be used to construct a static algorithm for the k-median problem with running time $n \cdot u(n, k) + q(n, k)$ by inserting each point in the input metric space U followed by a solution query. Hence, by Theorem D.1, we must have that $n \cdot u(n, k) + q(n, k) = \Omega(nk)$. Now assume that some incremental algorithm for k-median has query time $\tilde{O}(\text{poly}(k))$. If this algorithm also has an amortized update time of $\tilde{o}(k)$, then for the range of values of k where $q(n, k) = \tilde{o}(nk)$, it follows that $\tilde{o}(nk)$ is $\Omega(nk)$, giving a contradiction. Hence, the amortized update time must be $\tilde{\Omega}(k)$ and Theorem D.2 follows.

Theorem D.2. *Any $O(1)$-approximate incremental algorithm for the k-median problem with $\tilde{O}(\mathrm{poly}(k))$ query time must have $\tilde{\Omega}(k)$ amortized update time.*

It follows that the update time of our algorithm is optimal up to polylogarithmic factors.

E Omitted experimental results.

E.1 Update time evaluation.

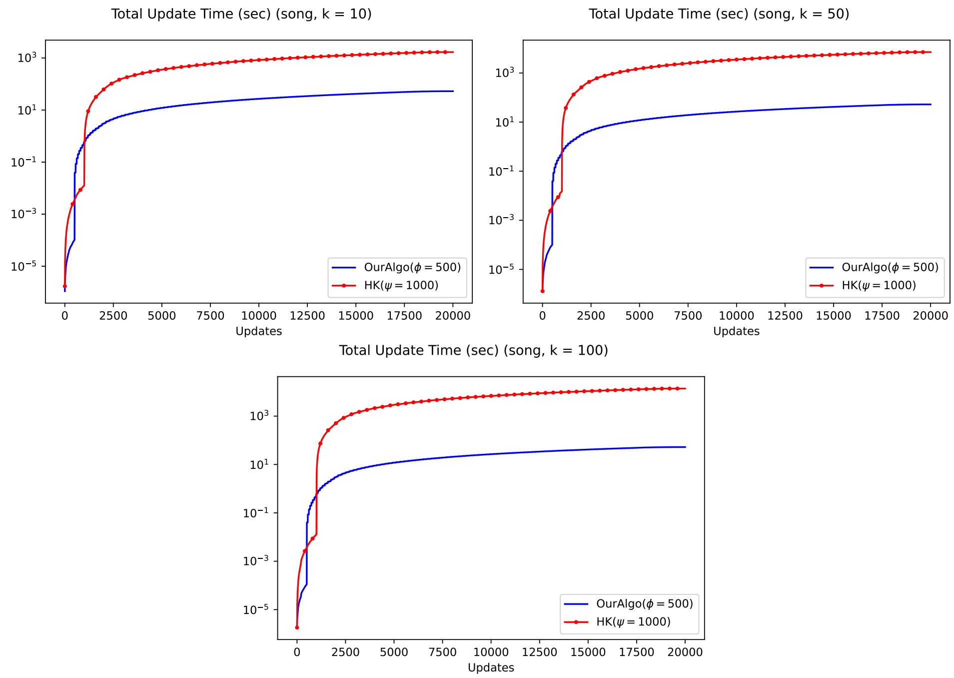

Figure 2: The cumulative update time for the different algorithms, on the Song dataset for $k = 10$ (top left), $k = 50$ (top right), $k = 100$ (bottom).

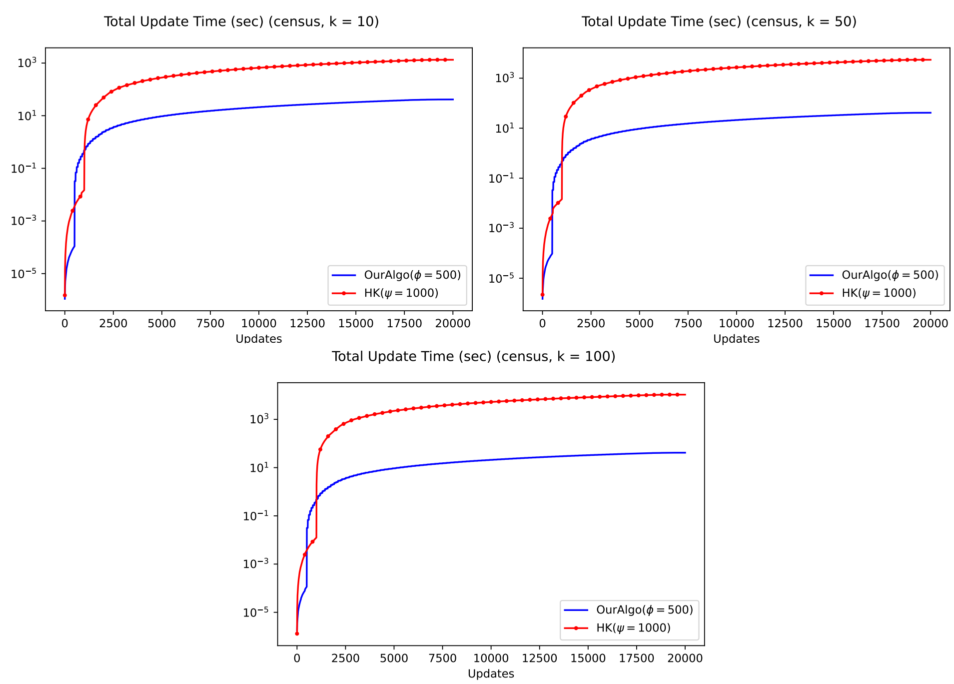

Figure 3: The cumulative update time for the different algorithms, on the Census dataset for $k = 10$ (top left), $k = 50$ (top right), $k = 100$ (bottom).

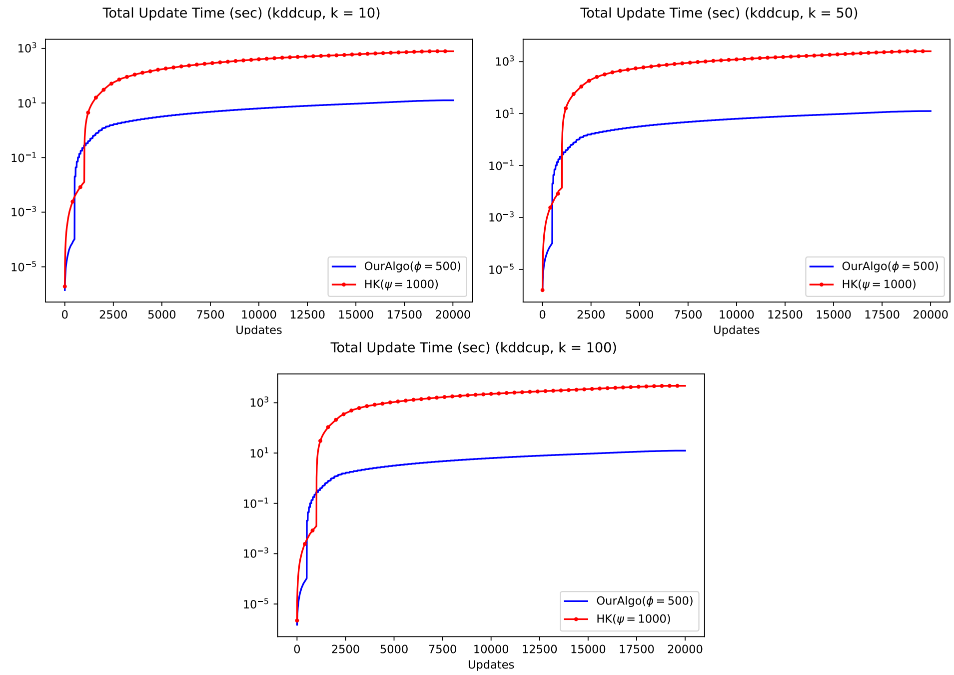

Figure 4: The cumulative update time for the different algorithms, on the KDD-Cup dataset for $k = 10$ (top left), $k = 50$ (top right), $k = 100$ (bottom).

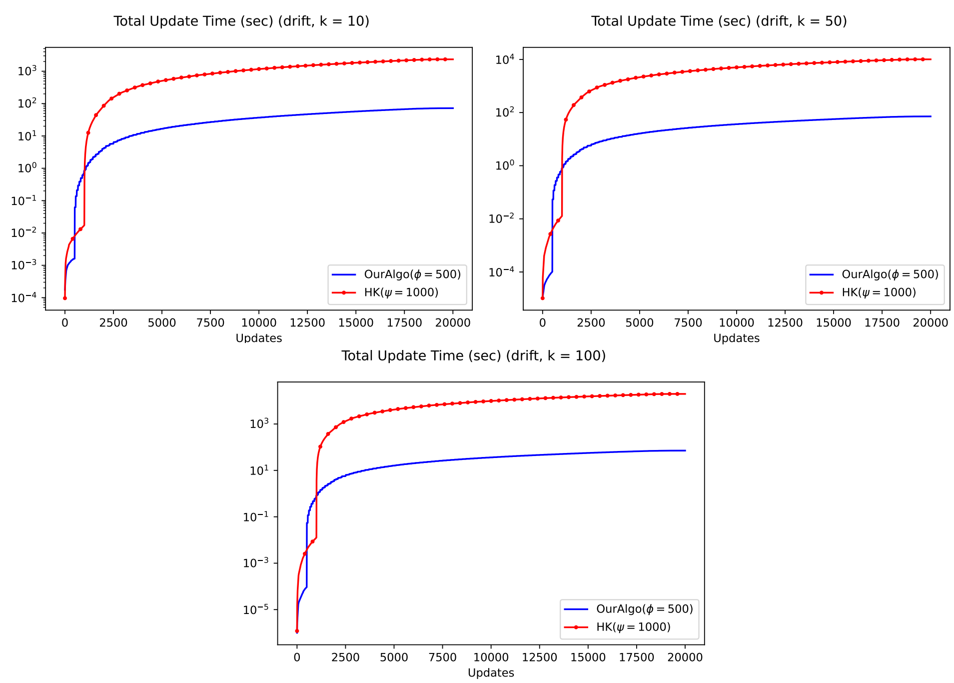

Figure 5: The cumulative update time for the different algorithms, on the Drift dataset for $k = 10$ (top left), $k = 50$ (top right), $k = 100$ (bottom).

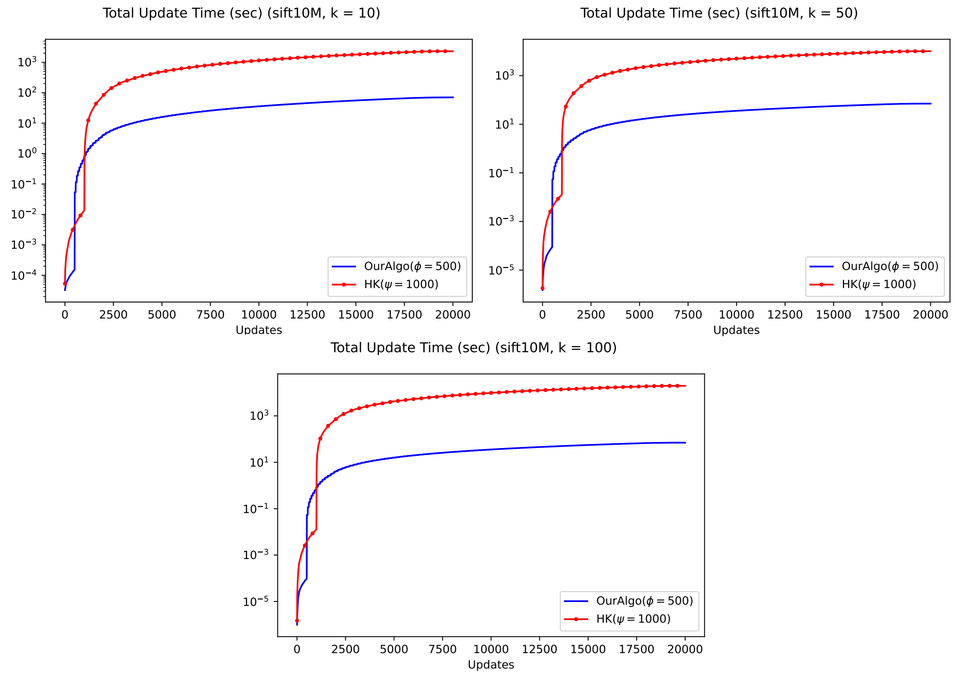

Figure 6: The cumulative update time for the different algorithms, on the SIFT10M dataset for $k = 10$ (top left), $k = 50$ (top right), $k = 100$ (bottom).

E.2 Solution cost evaluation.

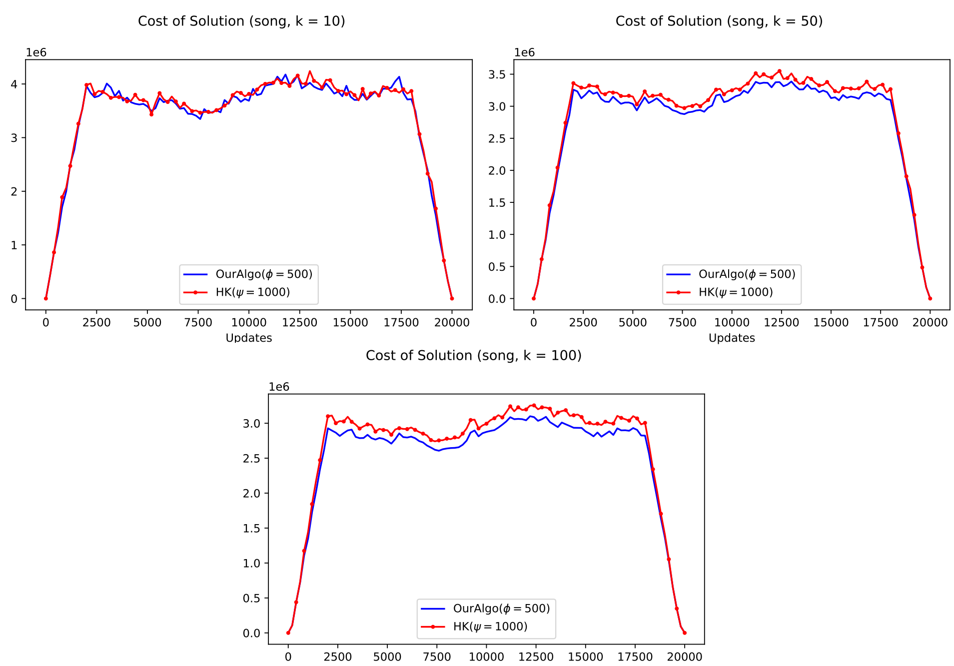

Figure 7: The solution cost by the different algorithms, on Song for $k = 10$ (top left), $k = 50$ (top right), $k = 100$ (bottom).

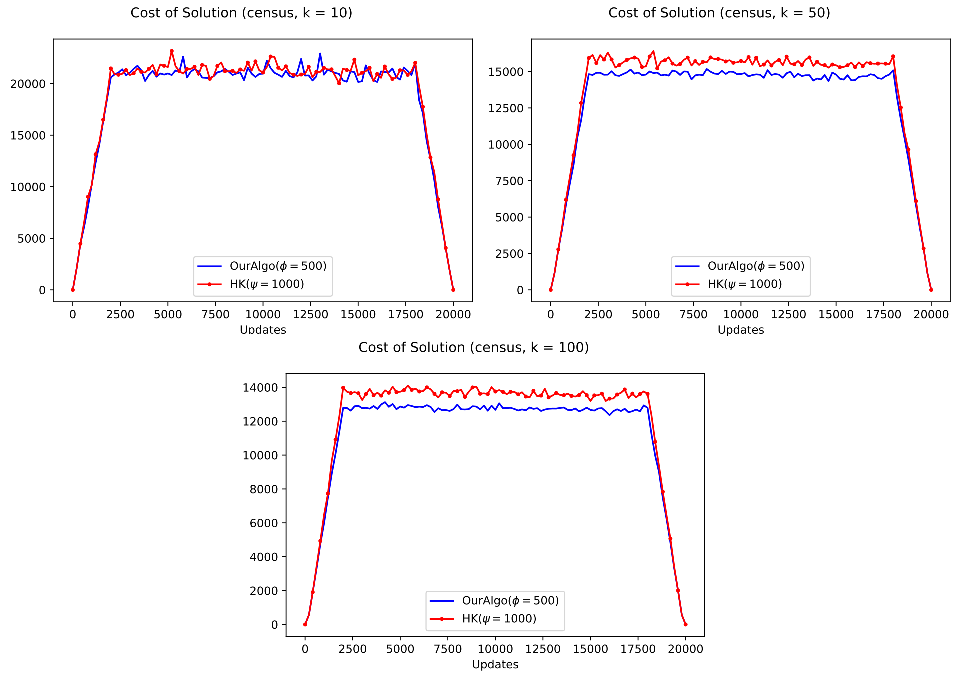

Figure 8: The solution cost by the different algorithms, on Census for $k = 10$ (top left), $k = 50$ (top right), $k = 100$ (bottom).

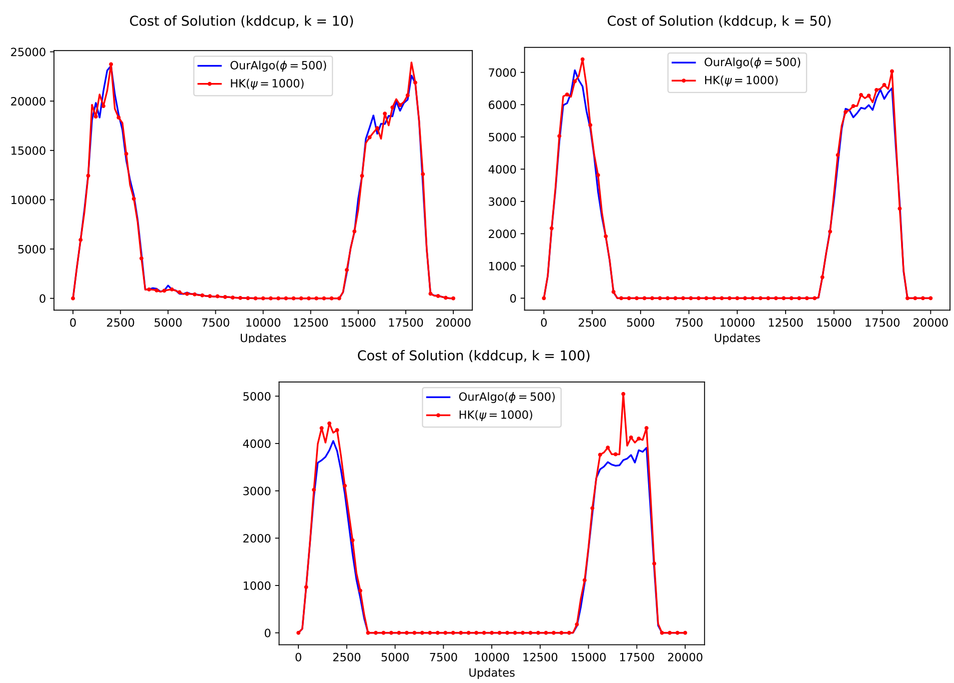

Figure 9: The solution cost by the different algorithms, on KDD-Cup for $k = 10$ (top left), $k = 50$ (top right), $k = 100$ (bottom).

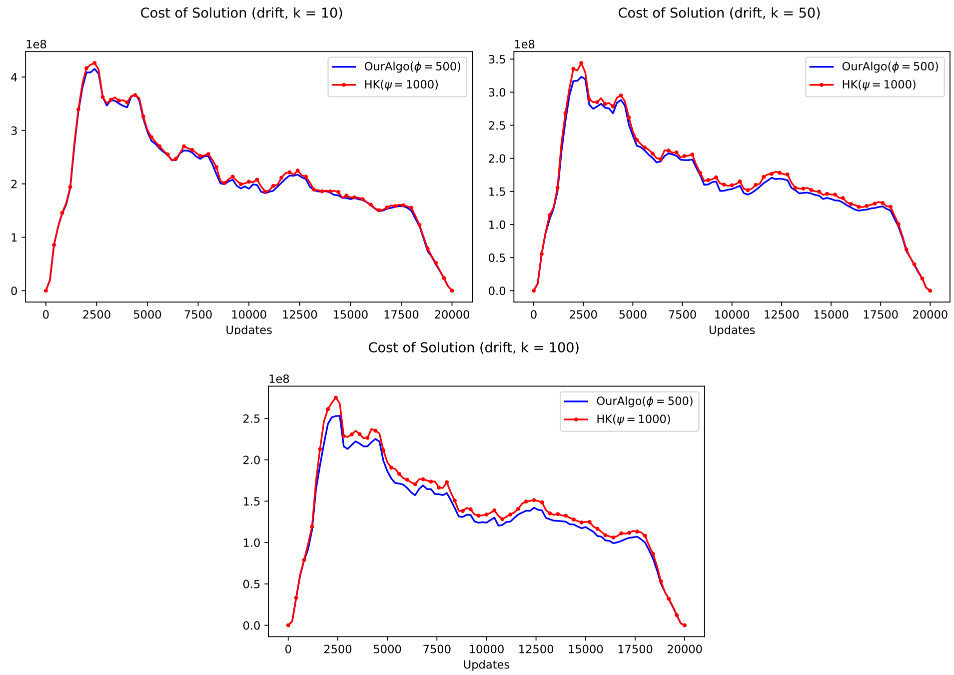

Figure 10: The solution cost by the different algorithms, on Drift for $k = 10$ (top left), $k = 50$ (top right), $k = 100$ (bottom).

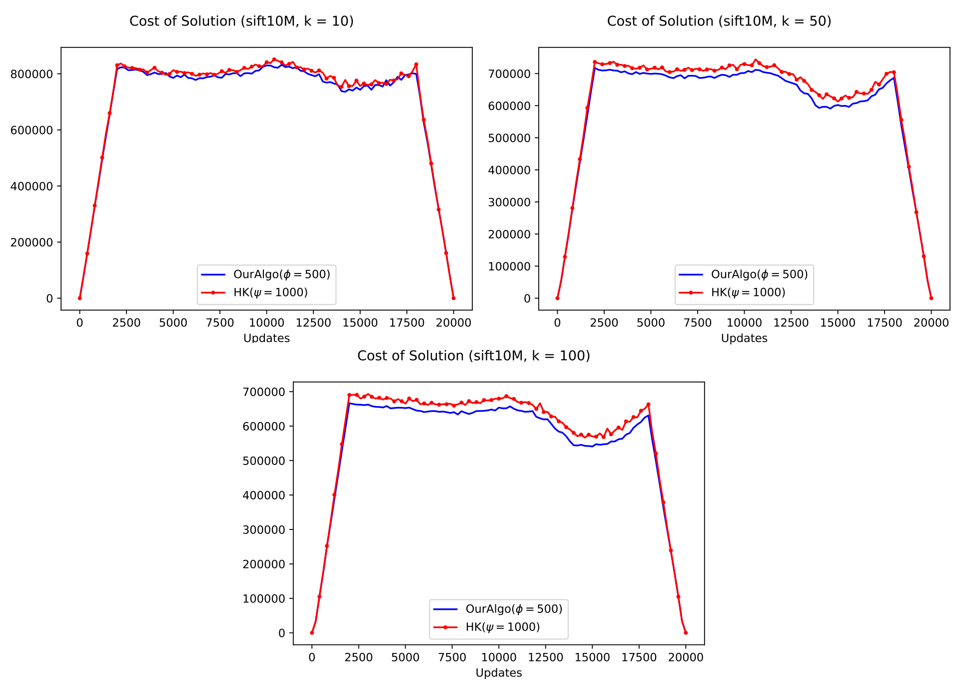

Figure 11: The solution cost by the different algorithms, on SIFT10M for $k = 10$ (top left), $k = 50$ (top right), $k = 100$ (bottom).

E.3 Query time evaluation

Table 4: The average query times for the algorithm OURALG($\phi = 500$) and HK($\psi = 1000$) (we omit the parameter value from the table to simplify the presentation), on the different datasets that we consider and for $k \in \{10, 50, 100\}$.

	Song		Census		KDD-Cup		Drift		SIFT10M	
	OURALG	HK	OURALG	HK	OURALG	HK	OURALG	HK	OURALG	HK
$k = 10$	0.569	0.327	0.478	0.280	0.069	0.176	0.729	0.421	0.732	0.419
$k = 50$	0.610	0.347	0.511	0.295	0.075	0.141	0.784	0.447	0.795	0.448
$k = 100$	0.665	0.373	0.552	0.317	0.085	0.131	0.857	0.483	0.866	0.483

E.4 Parameter tuning.

E.4.1 Update time.

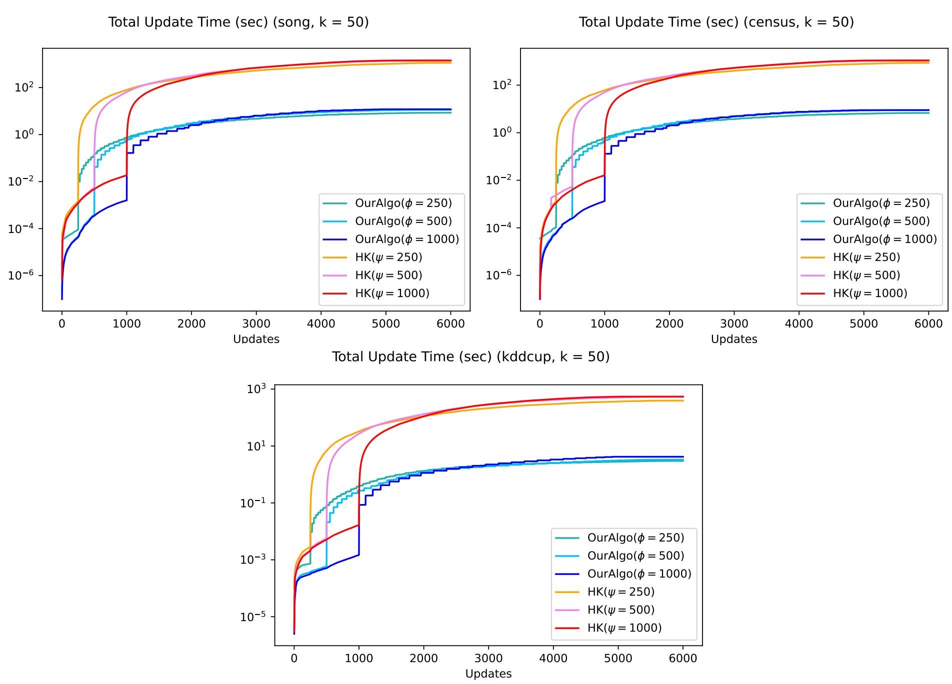

Figure 12: The cumulative update time for different parameters of OURALG and HK, for $k = 50$, on datasets Song (top left), Census (top right), and KDD-Cup (bottom).

E.4.2 Solution cost.

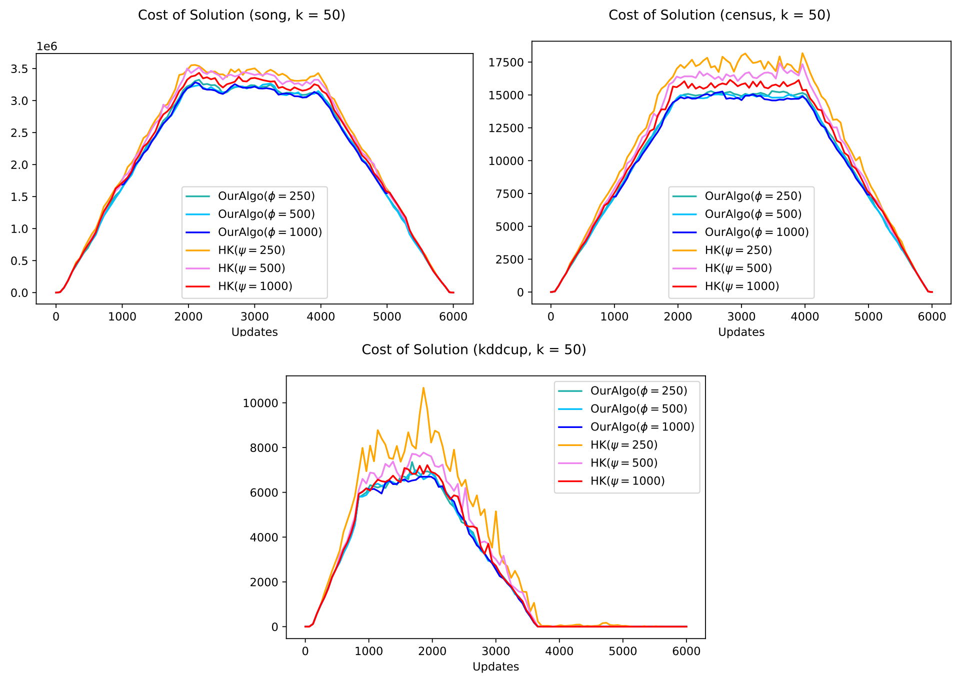

Figure 13: The solution cost for different parameters of OURALG and HK, for $k = 50$, on datasets Song (top left), Census (top right), and KDD-Cup (bottom).

E.4.3 Query time.

Table 5: The average query times for the algorithm OURALG and HK with different parameters, on the different datasets for $k = 50$.

	Song	Census	KDD-Cup
HK($\psi = 250$)	0.026	0.021	0.012
HK($\psi = 500$)	0.087	0.073	0.043
HK($\psi = 1000$)	0.293	0.249	0.156
OURALG($\phi = 250$)	0.223	0.187	0.054
OURALG($\phi = 500$)	0.439	0.364	0.086
OURALG($\phi = 1000$)	0.719	0.605	0.146

E.5 Randomized order of updates.

E.5.1 Update time.

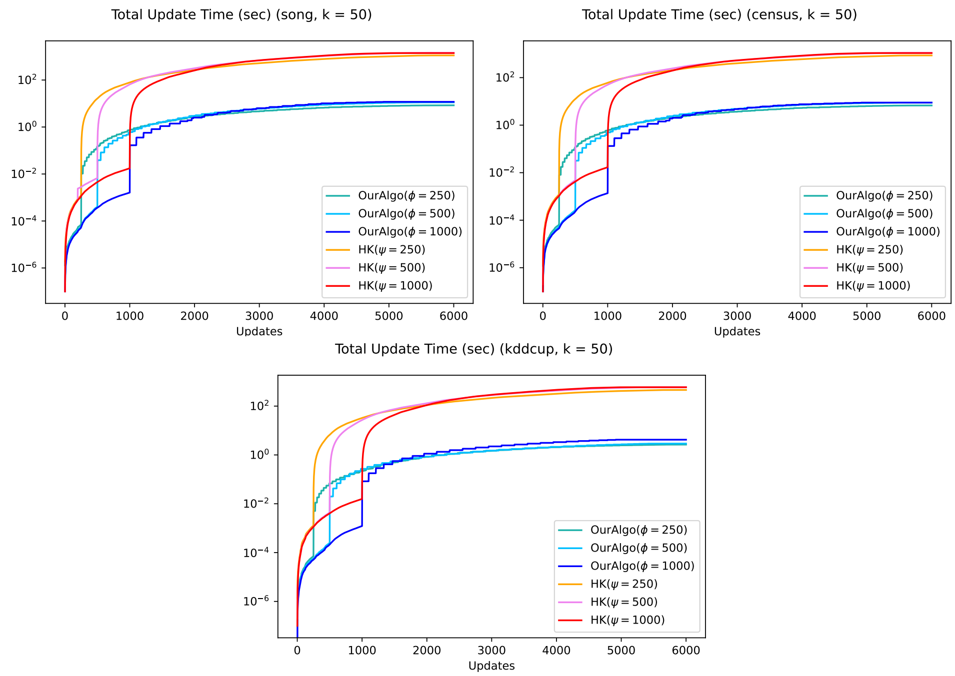

Figure 14: The cumulative update time for different parameters of OURALG and HK, for $k = 50$, over a sequence of updates given by a randomized order of the points in the dataset, on the datasets Song (top left), Census (top right), and KDD-Cup (bottom).

E.5.2 Solution cost.

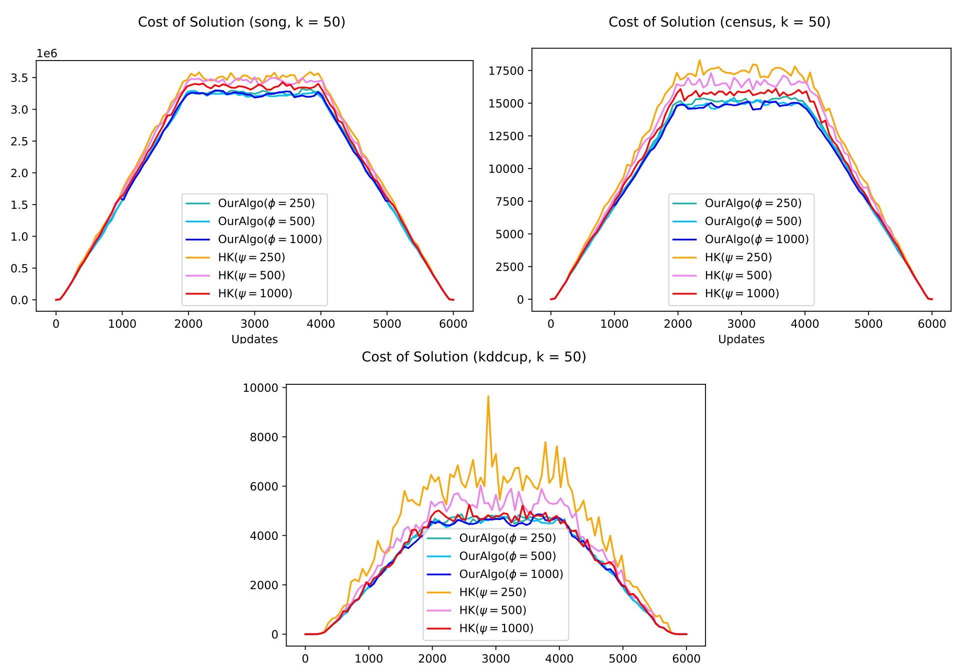

Figure 15: The solution cost for different parameters of OURALG and HK, for $k = 50$, over a sequence of updates given by a randomized order of the points in the dataset, on the datasets Song (top left), Census (top right), and KDD-Cup (bottom).

E.5.3 Query time.

Table 6: The average query times for the algorithm OURALG and HK with different parameters, for $k = 50$, over a sequence of updates given by a randomized order of the points in each of the datasets that we consider.

	Song	Census	KDD-Cup
HK($\psi = 250$)	0.025	0.021	0.014
HK($\psi = 500$)	0.086	0.073	0.050
HK($\psi = 1000$)	0.292	0.247	0.173
OURALG($\phi = 250$)	0.225	0.185	0.062
OURALG($\phi = 500$)	0.440	0.364	0.100
OURALG($\phi = 1000$)	0.723	0.605	0.165

E.6 Larger experiment.

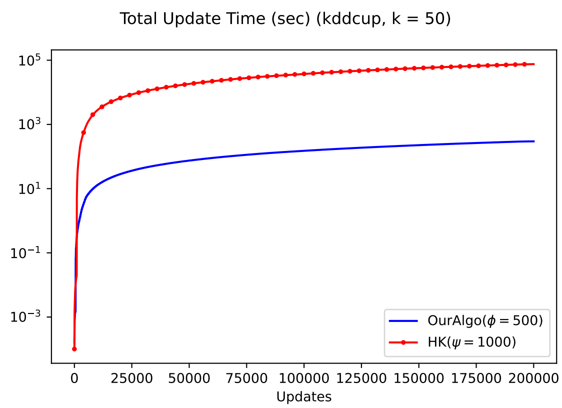

Figure 16: The total update time for OURALG($\phi = 500$) and HK($\psi = 1000$), on the larger instance derived from KDD-Cup, for $k = 50$.

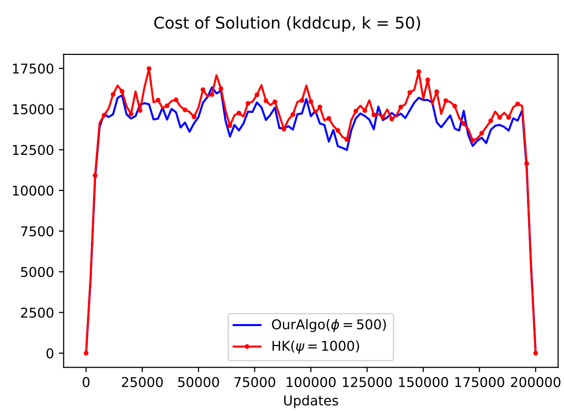

Figure 17: The solution cost produced by OURALG($\phi = 500$) and HK($\psi = 1000$) two algorithms, on the larger instance derived from KDD-Cup, for $k = 50$.

The average query times for OURALG($\phi = 500$) and HK($\psi = 1000$) while handling this longer sequence of updates were 0.416 and 0.225 respectively.