JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 40, 245-251 (2024)
DOI: 10.6688/JISE.202403_40(2).0003

Enumerating Furthest Pairs in Ultrametric Spaces

HUI-TING CHEN AND CHING-LUEH CHANG+
Department of Computer Science and Engineering
Yuan Ze University
Taoyuan City, 320315 Taiwan
E-mail: s1049102@mail.yzu.edu.tw; clchang@saturn.yzu.edu.tw+

We prove the following results on enumerating or counting furthest pairs given an ultrametric space with n elements:

- There is a deterministic $O(F + n\log n)$-time algorithm for enumerating all furthest pairs, where F denotes the total number of furthest pairs.
- There is a Monte Carlo $O(n/\varepsilon^2)$-time algorithm that estimates the number of furthest pairs to within a multiplicative factor in $(1-\varepsilon, 1+\varepsilon)$, where $\varepsilon > 0$. Furthermore, the time complexity of $O(n/\varepsilon^2)$ cannot be improved to $o(n \cdot f(\varepsilon))$ for any $f(\cdot)$.

Keywords: ultrametric space, furthest pairs, combinatorial enumeration, counting, Monte Carlo algorithm

1. INTRODUCTION

An ultrametric space is a nonempty set M endowed with $d\colon M \times M \to [0,\infty)$ such that

- $d(x,y) = 0$ iff $x = y$ (identity of indiscernibles),
- $d(x,y) = d(y,x)$ (symmetry), and
- $d(x,z) \le \max\{d(x,y), d(y,z)\}$ (strong triangle inequality)

for all x, y, $z \in M$. It is fundamental in mathematical analysis.

Consider the problem of enumerating/counting point pairs with the longest distance (called the diameter) in an n-point ultrametric space. The problem can be solved trivially in $O(n^2)$. We show the following:

- There is a deterministic $O(F + n\log n)$-time algorithm for enumerating all furthest pairs, where F denotes the total number of furthest pairs. (A pair $(a,b) \in M^2$ is furthest if $d(a,b)$ is the diameter.)
- There is a Monte Carlo $O(n/\varepsilon^2)$-time algorithm that estimates the number of furthest pairs to within a multiplicative factor in $(1-\varepsilon, 1+\varepsilon)$, where $\varepsilon > 0$. Furthermore, the time complexity of $O(n/\varepsilon^2)$ cannot be improved to $o(n \cdot f(\varepsilon))$ for any $f(\cdot)$.

Received September 26, 2022; revised March 25, 2023; accepted May 15, 2023.
Communicated by Jia-Ming Chang.
+ Corresponding author.

Input: Nonempty $S \subseteq [n]$
1: Pick $p \in S$ arbitrarily;
2: **for all** $s \in S$ **do**
3: Query for $d(p,s)$;
4: **if** $d(p,s) = \Delta$ **then**
5: Print (p,s);
6: **end if**
7: **end for**
8: $T \leftarrow \{s \in S \setminus \{p\} \mid d(p,s) = \Delta\}$;
9: Print all pairs in $T \times (S \setminus (T \cup \{p\}))$;
10: **if** $T \neq \emptyset$ **then**
11: Enum.-furthest(T);
12: **end if**

Fig. 1. Algorithm Enum.-furthest for enumerating all $(s,s') \in S^2$ satisfying $d(s,s') = \Delta$, where Δ is obtained by Furthest-Pair during preprocessing (done only once).

Clearly, no $o(F)$-time algorithms can enumerate all F furthest pairs. So our first algorithm is optimal up to an additive $O(n \log n)$. Our second algorithm takes $o(n^2)$ time; hence it only reads an $o(1)$ proportion of distances.

The problem of finding all furthest pairs on the Euclidean plane can be solved in $O(n \log n)$ time [1]. Another problem is to compute, for each vertex p_1 of a simple polygon P, a vertex p_2 of P with the maximum geodesic distance to p_1, where the geodesic distance between p_1 and p_2 is the minimum distance needed to go from p_1 to p_2 along the boundary of P. This problem has an $O(n \log n)$-time $O(n)$-space algorithm [2]. In recent years, there are a lot researches on $o(n^2)$-time algorithms for metric-space problems, especially in big data. Usually, we just get approximate answers. There is a lot of algorithmic research along these lines. This includes research on approximate furthest pairs in metric spaces [3] and an $O(1/\varepsilon^{O(1)})$-time $(1+\varepsilon)$-approximation algorithm for the 1-median problem in ultrametric spaces [4]. In [5], a heuristic is designed to find the furthest neighbor of a given point. All known algorithms for finding a furthest pair among n points in $\mathbb{R}^d$ require $\Omega(n^{2-1/\Theta(d)})$ time [6].

2. ENUMERATING ALL FURTHEST PAIRS

Let $([n],d)$ be an ultrametric space with diameter $\Delta \equiv \max_{x,y \in [n]} d(x,y)$ and $\varepsilon > 0$, where $[n] \equiv \{1,2,\ldots,n\}$. It is well-known that there exists a deterministic $O(n)$-time algorithm, hereafter called Furthest-Pair, for finding $(a,b) \in [n]^2$ satisfying $d(a,b) = \Delta$. Assume all pairs in $[n]^2$ to be unordered. For example, $[2] \times ([n] \setminus [2])$ contains $2(n-2)$ (rather than $4(n-2)$) pairs. Define

$$F \equiv \left|\left\{(u,v) \in [n]^2 \mid d(u,v) = \Delta\right\}\right| \tag{1}$$

to be the number of furthest pairs.

Lemma 1. *In Algorithm* Enum.-furthest *(line 9, in Fig. 1), each* $(s,s') \in T \times (S \setminus (T \cup \{p\}))$ *satisfies* $d(s,s') = \Delta$.

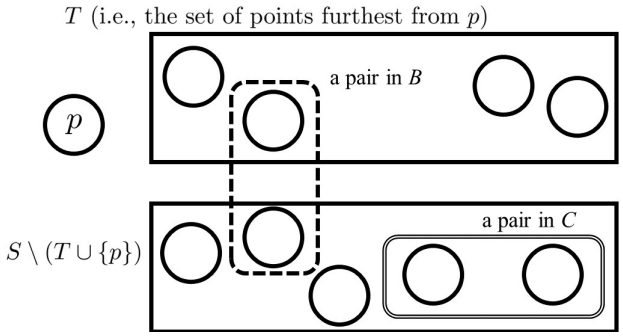


Fig. 2. An illustration of p (the leftmost circle), T (the upper rectangle), $S \setminus (T \cup \{p\})$ (the lower rectangle), a pair in B (the dashed rounded rectangle) and a pair in C (the double rounded rectangle).

Proof. Take any $s \in T$ and $s' \in S \setminus (T \cup \{p\})$. By the strong triangle inequality, $d(p,s) \leq \max\{d(p,s'), d(s',s)\}$. By line 8, $d(p,s) = \Delta$ and $d(p,s') < \Delta$. So $d(s,s') = \Delta$. □

Lemma 2. *In Algorithm* Enum.-furthest *(in Fig. 1), each* $(t,t') \in (S \setminus (T \cup \{p\}))^2$ *satisfies* $d(t,t') < \Delta$.

Proof. By the strong triangle inequality, $d(t,t') \leq \max\{d(p,t), d(p,t')\}$. As t, $t' \in S \setminus (T \cup \{p\})$, we have $d(p,t)$, $d(p,t') < \Delta$ by line 8. □

Lemma 3. *Algorithm* Enum.-furthest *(in Fig. 1) enumerates all* $(s,s') \in S^2$ *satisfying* $d(s,s') = \Delta$ *(and nothing else).*

Proof. The lemma is trivial when $|S| \leq 1$. Assume as induction hypothesis that the recursive call in line 11 outputs all $(s,s') \in T^2$ satisfying $d(s,s') = \Delta$ (and nothing else). Clearly, lines 2–7 output all pairs in $\{p\} \times S$ with distance Δ. The set of pairs in S^2 but not in $\{p\} \times S$ is exactly

$$(S \setminus \{p\})^2 = T^2 \cup B \cup C, \tag{2}$$

where

$$B = T \times (S \setminus (T \cup \{p\})),$$
$$C = (S \setminus (T \cup \{p\}))^2.$$

See Fig. 2 for an illustration.[1] The induction hypothesis says that all pairs in T^2 with distance Δ are printed by the recursive call in line 11. By Lemma 1, all pairs in B have distance Δ – These are exactly the outputs of line 9. By Lemma 2, each pair in C has a distance less than Δ. Clearly, no pairs in C are printed. □

Enum.-furthest$([n])$ makes several levels of recursive calls.[2] We say that a recursive call is bad if $|T| < |S|/2$ (where T is as in line 8) and good otherwise.

[1] Recall that pairs are unordered by default in this paper.

[2] Enum.-furthest$([n])$ denotes Enum.-furthest with the whole ground-set $[n]$ as input.

Lemma 4. *There are at most* $\lg n$ *bad recursive calls.*

Proof. A bad recursive call at least halves the size of the argument to Enum.-furthest (from S to T in line 11, where $|T| < |S|/2$ by badness). □

Lemma 5. *In a good recursive call, lines 1-9 take time at most proportional to the number of pairs printed.*

Proof. Clearly, lines 2-7 and 9 print exactly $|T|$ and $|T \times (S \setminus (T \cup \{p\}))|$ pairs, respectively. So the number of pairs printed is $|T| + |T \times (S \setminus (T \cup \{p\}))|$. Clearly, lines 1-9 take time $O(|S| + |T \times (S \setminus (T \cup \{p\}))|)$. By goodness, $|T| \geq |S|/2$. *I.e.*, $|S| \leq 2|T|$. □

Lemma 6. *In a bad recursive call, lines 1-9 take time* $O(|S|)$ *plus a quantity at most proportional to the number of pairs printed.*

Proof. Clearly, lines 1-8 and 9 take time $O(|S|)$ and $O(|T \times (S \setminus (T \cup \{p\}))|)$, respectively. Line 9 alone prints $|T \times (S \setminus (T \cup \{p\}))|$ pairs. □

Recall that F is the number of furthest pairs. We now prove that Enum.-furthest$([n])$ enumerates furthest pairs in $O(F + n \log n)$ time. Clearly, writing down all F furthest pairs takes time $\Omega(F)$. So our algorithm is optimal up to an additive $O(n \log n)$.

Theorem 7. Enum.-furthest$([n])$ *takes* $O(F + n \log n)$ *time and enumerates all pairs* $(s, s') \in [n]^2$ *satisfying* $d(s, s') = \Delta$.

Proof. By Lemma 3, Enum.-furthest$([n])$ enumerates all F furthest pairs (and nothing else). By Lemma 5, the time taken by the good recursive calls is at most proportional to the total number of pairs printed, or F. By Lemma 6, bad recursive calls take a total of

$$O\left(\sum_{i=1}^{k} |S_i|\right) + O(F) \tag{3}$$

time, where k denotes the number of bad recursive calls and S_i the argument to the ith bad recursive call. By Lemma 4, there are at most $\lg n$ bad recursive calls, *i.e.*, $k \leq \lg n$. So $\sum_{i=1}^{k} |S_i| = O(n \log n)$. □

3. RANDOMIZED COUNTING

Theorem 8 ([7])**.** *(Chernoff's Bounds). Let* $X = \sum_{i=1}^{n} X_i$*, where* $X_i = 1$ *with probability* p *and* $X_i = 0$ *with probability* $1 - p$*, and all* X_i *are independent. Let* $\mu = \mathbb{E}(X) = np$*. Then*

$$\Pr[X \geq (1+\delta)\mu] \leq e^{-\frac{\delta^2}{2+\delta}\mu},$$
$$\Pr[X \leq (1-\delta)\mu] \leq e^{-\mu\delta^2/2}$$

for all $0 < \delta < 1$.

Lemma 9. $F \geq n$.

Proof. Let $a, b \in [n]$ be such that $d(a,b) = \Delta$. By the strong triangle inequality, either $d(a,x)$ or $d(b,x)$ (or both) equals Δ for each $x \in [n]$. □

By convention, a Monte Carlo algorithm is allowed to err with probability $1/3$ (or any small constant).

Theorem 10. *There exists a Monte Carlo $O(n/\varepsilon^2)$-time algorithm estimating F to within a multiplicative factor in $(1-\varepsilon, 1+\varepsilon)$, for all $\varepsilon > 0$.*

Proof. Take m independent and uniformly random pairs, $\{(a_i,b_i) \in [n]^2\}_{i=1}^m$, for m to be determined later. So a_i and b_i are uniformly random elements of an ultrametric space $([n],d)$ for all $1 \le i \le m$. In expectation, $\{(a_i,b_i)\}_{i=1}^m$ contains $mF/\binom{n}{2}$ furthest pairs. By Chernoff's bound, $\{(a_i,b_i)\}_{i=1}^m$ contains more than $(1+\varepsilon)mF/\binom{n}{2}$ or fewer than $(1-\varepsilon)mF/\binom{n}{2}$ furthest pairs with probability $\exp(-\Omega(\varepsilon^2 mF/\binom{n}{2}))$. By Lemma 9, $\exp(-\Omega(\varepsilon^2 mF/\binom{n}{2})) \le \exp(-\Omega(\varepsilon^2 m/n)$. Taking $m = Cn/\varepsilon^2$ for a sufficiently large constant $C > 0$ drives the error probability below $1/3$. □

An immediate question is: Can the time complexity in Theorem 10 be improved to $o(n \cdot f(\varepsilon))$ for some $f(\cdot)$? The answer is negative.

Theorem 11. *There does not exist a Monte Carlo $o(n)$-time algorithm estimating F to within a multiplicative factor in $[1/C, C]$, for any constant $C > 1$.*

Proof. Consider ultrametric spaces $([n],d)$ such that there exists a set $S \subseteq [n]$ satisfying (1) $d(s,x) = \Delta \gg 1$ for all $s \in S$ and $x \in [n] \setminus \{s\}$, and (2) $d(x,y) = 1$ for all distinct x, $y \in [n] \setminus S$. Let $B > C^{100}$ be any large constant. Then pick u_1, u_2, …, u_B independently and uniformly at random from $[n]$. Consider the following cases:

Case 1: $S = \{u_1, u_2, \ldots, u_B\}$. So about Bn distances are furthest.

Case 2: $S = \{u_1\}$. So about n distances are furthest.

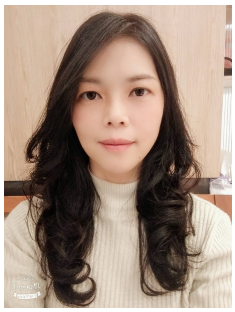
With $o(n)$ queries, the probability of obtaining a non-1 distance is $o(1)$ in both cases. So with probability $1 - o(1)$, it will be information-theoretically impossible to distinguish between the two cases. If F can be approximated to within a multiplicative factor in $[1/C, C]$, then we should be able to distinguish between the two cases, a contradiction. □

4. CONCLUSION

Consider the problem of enumerating/counting point pairs with the longest distance (called the diameter) in an n-point ultrametric space. We give a deterministic $O(F + n\log n)$-time algorithm for enumerating all furthest pairs, where F denotes the total number of furthest pairs. Then we give a Monte Carlo $O(n/\varepsilon^2)$-time algorithm estimating F to within a multiplicative factor in $(1-\varepsilon, 1+\varepsilon)$, for all $\varepsilon > 0$. Finally, we prove the non-existence of a Monte Carlo $o(n)$-time algorithm estimating F to within a multiplicative factor in $[1/C, C]$, for any constant $C > 1$.

REFERENCES

1. B. K. Bhattacharya and G. T. Toussaint, "On geometric algorithms that use the furthest-point voronoi diagram," *Machine Intelligence and Pattern Recognition*, Elsevier, 1985, Vol. 2, pp. 43-61.
2. S. Suri, "Computing geodesic furthest neighbors in simple polygons," *Journal of Computer and System Sciences*, Vol. 39, 1989, pp. 220-235.
3. P. Indyk, "Sublinear time algorithms for metric space problems," in *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, 1999, pp. 428-434.
4. C.-L. Chang, "On ultrametric 1-median selection," *Theoretical Computer Science*, Vol. 828-829, 2020, pp. 65-69.
5. A. S. Tarawneh, A. B. Hassanat, I. Elkhadiri, D. Chetverikov, and M. Alrashidi, "K-means tree for fast furthest neighbor approximation," in *Proceedings of the 16th IEEE International Computer Engineering Conference*, 2020, pp. 77-82.
6. R. Williams, "On the difference between closest, furthest, and orthogonal pairs: Nearly-linear vs barely-subquadratic complexity," in *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2018, pp. 1207-1215.
7. T. Hagerup and C. Rüb, "A guided tour of Chernoff bounds," *Information Processing Letters*, Vol. 33, 1990, pp. 305-308.



Hui-Ting Chen was born in Zhubei City, Hsinchu County, Taiwan in 1979. She received the BS and MS degrees in Computer Science and Information Engineering from National Defense University in 2001 and 2005. She is currently pursuing the Ph.D. degree in Computer Science at Yuan Ze University, Taoyuan, Taiwan. Her research interest includes the graph theory and theoretical computer science.

Ching-Lueh Chang received his BS, MS and Ph.D. degrees in 2004, 2006 and 2010, respectively, from National Taiwan University, Taipei, Taiwan. He was an Assistant Professor of Yuan Ze University, Taoyuan, Taiwan, from August 2010 to July 2013. Since August 2013, he has been an Associate Professor of YZU. He is interested in theoretical computer science. In recent years, he has been studying the 1-median problem in metric spaces. In particular, he has characterized the optimal approximation ratios achievable by deterministic $O(n^c)$-time algorithms for *all* constants $c > 1$. He has published in *Theoretical Computer Science*, *Theory of Computing Systems*, *Information Processing Letters*, *Discrete Applied Mathematics*, *International Journal of Foundations of Computer Science*, *Journal of Computer and System Sciences* and *ACM Transactions on Computation Theory*.