Encyclopedia of Algorithms
DOI 10.1007/978-3-642-27848-8_568-1

Approximate Distance Oracles with Improved Query Time

Christian Wulff-Nilsen*
Department of Computer Science, University of Copenhagen, Copenhagen, Denmark

Keywords Approximate distance oracle • Shortest paths • Graphs • Query time

Years and Authors of Summarized Original Work

2013; Wulff-Nilsen

Problem Definition

This problem is concerned with obtaining a compact data structure capable of efficiently reporting approximate shortest path distance queries in a given undirected edge-weighted graph $G = (V, E)$. If the query time is independent (or nearly independent) of the size of G, we refer to the data structure as an *approximate distance oracle* for G. For vertices u and v in G, we denote by $d_G(u, v)$ the shortest path distance between u and v in G. For a given *stretch* parameter $\delta \geq 1$, we call the oracle *δ-approximate* if for all vertices u and v in G, $d_G(u, v) \leq \tilde{d}_G(u, v) \leq \delta d_G(u, v)$, where $\tilde{d}_G(u, v)$ is the output of the query for u and v. Hence, we allow estimates to be stretched by a factor up to δ but not shrunk.

Key Results

A major result in the area of distance oracles is due to Thorup and Zwick [4]. They gave, for every integer $k \geq 1$, a $(2k - 1)$-approximate distance oracle of size $O\left(kn^{1+1/k}\right)$ and query time $O(k)$, where n is the number of vertices of the graph. This is constant query time when k is constant. Corresponding approximate shortest paths can be reported in time proportional to their length. Mendel and Naor [3] asked the question of whether query time can be improved to a universal constant (independent also of k) while keeping both size and stretch small. They obtained $O\left(n^{1+1/k}\right)$ size and $O(1)$ query time at the cost of a constant-factor increase in stretch to $128k$. Unlike the oracle of Thorup and Zwick, Mendel and Naor's oracle is not path reporting.

In [5], it is shown how to improve the query time of Thorup-Zwick to $O(\log k)$ without increasing space or stretch. This is done while keeping essentially the same data structure but applying binary instead of linear search in so-called bunch structures that were introduced by Thorup and Zwick [4]; the formal definition will be given below. Furthermore, it is shown in [5] how to improve the stretch of the oracle of Mendel and Naor to $(2 + \epsilon)k$ for an arbitrarily small constant $\epsilon > 0$ while keeping query time constant (bounded by $1/\epsilon$). This improvement is obtained without an increase in space except for large values of k close to $\log n$ (only values of k less than

*E-mail: koolooz@di.ku.dk

DOI 10.1007/978-3-642-27848-8_568-1

$\log n$ are interesting since the Mendel-Naor oracle has optimal $O(n)$ space and $O(1)$ query time for larger values). Below, we sketch the main ideas in the improvement of Thorup-Zwick and of Mendel-Naor, respectively.

Oracle with $O(\log k)$ Query Time

The oracle of Thorup and Zwick keeps a hierarchy of sets of sampled vertices $V = A_0 \supseteq A_1 \supseteq A_2 \ldots \supseteq A_k = \emptyset$, where for $i = 1, \ldots, k-1$, A_i is obtained by picking each element of A_{i-1} independently with probability $n^{-1/k}$. Define $p_i(u)$ as the vertex in A_i closest to u. The oracle precomputes and stores for each vertex $u \in V$ the *bunch* B_u, defined as

$$B_u = \bigcup_{i=0}^{k-1} \{v \in A_i \setminus A_{i+1} | d_G(u, v) < d_G(u, p_{i+1}(u))\}.$$

See Fig. 1 for an illustration of a bunch. The distance $d_G(u, v)$ for each $v \in B_u$ is precomputed as well.

Now, to answer a query for a vertex pair (u, v), the oracle performs a linear search through bunches B_u and B_v. Pseudocode is given in Fig. 2. It is clear that query time is $O(k)$, and it can be shown that the estimate output in line 6 has stretch $2k - 1$.

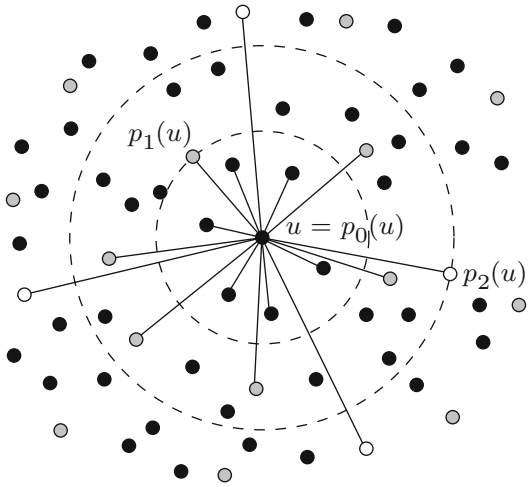

Fig. 1 A bunch B_u in a complete Euclidean graph with $k = 3$. *Black vertices* belong to A_0, *grey vertices* to A_1, and *white vertices* to A_2. Line segments connect u to vertices of B_u

```
Algorithm dist_k(u, v)
1. w ← p_0(u); j ←0
2. while w ∉ B_v
3.     j ← j + 1
4.     (u, v) ← (v, u)
5.     w ← p_j(u)
6. return d_G(w, u) + d_G(w, v)
```

Fig. 2 Answering a distance query, starting at sample level i

DOI 10.1007/978-3-642-27848-8_568-1

We can improve query time to $O(\log k)$ by instead doing binary search in the bunch structures. A crucial property of the Thorup-Zwick oracle is that every time the test in line 2 succeeds, $d_G(u, p_j(u))$ increases by at most $d_G(u, v)$, and this is sufficient to prove $2k - 1$ stretch. In particular, if the test succeeds two times in a row, $d_G(u, p_{j+2}(u)) - d_G(u, p_j(u)) \leq 2d_G(u, v)$, where j is even. If we can check that $d_G(u, p_{j'+2}(u)) - d_G(u, p_{j'}(u)) \leq 2d_G(u, v)$ for all smaller even indices j', we may start the query algorithm at index j instead of index 0. Since we would like to apply binary search, pick j to be (roughly) $k/2$. It suffices to check only one inequality, namely, the one with the largest value $d_G(u, p_{j'+2}(u)) - d_G(u, p_{j'}(u))$. Note that this value depends only on u and k, so we can precompute the index j' with this largest value. In the query phase, we can check in $O(1)$ time whether $d_G(u, p_{j'+2}(u)) - d_G(u, p_{j'}(u)) \leq 2d_G(u, v)$. If the test succeeds, we can start the query at j, and hence, we can recurse on indices between j and $k - 1$. Conversely, if the test fails, it means that the test in line 2 fails for either j' or $j' + 1$. Hence, the query algorithm of Thorup and Zwick terminates no later than at index $j' + 1$, and we can recurse on indices between 0 and $j' + 1$. In both cases, the number of remaining indices is reduced by a factor of at least 2. Since each recursive call takes $O(1)$ time, we thus achieve $O(\log k)$ query time.

Since the improved oracle is very similar to the Thorup-Zwick oracle, it is path reporting, i.e., it can report approximate paths in time proportional to their length.

Oracle with Constant Query Time

The second oracle in [5] can be viewed as a hybrid between the oracles of Thorup-Zwick and of Mendel-Naor. An initial estimate is obtained by querying the Mendel-Naor oracle. This estimate has stretch at most $128k$, and it is refined in subsequent iterations until the desired stretch $(2 + \epsilon)k$ is obtained. In each iteration, the current estimate is reduced by a small constant factor greater than 1 (depending on ϵ). Note that after a constant number of iterations, the estimate will be below the desired stretch, but it needs to be ensured that it is not below the shortest path distance.

In each iteration, the hybrid algorithm attempts to start the Thorup-Zwick query algorithm at a step corresponding to this estimate. If this can be achieved, only a constant number of steps of this query algorithm need to be executed before the desired stretch is obtained. Conversely, if the hybrid algorithm fails to access the Thorup-Zwick oracle in any iteration, then by a property of the bunch structures, it is shown that the current estimate is not below the shortest path distance. Hence, the desired stretch is again obtained.

An important property of the Mendel-Naor oracle needed above is that the set d_{MN} of all different values the oracle can output has size bounded by $O\left(n^{1+1/k}\right)$. This is used in the hybrid algorithm as follows. In a preprocessing step, values from d_{MN} are ordered in a list $\mathcal{L}$ together with additional values corresponding to the intermediate estimates that the hybrid algorithm can consider in an iteration. Updating the estimate in each iteration then corresponds to a linear traversal of part of $\mathcal{L}$. Next, each vertex p_i of each bunch structure B_u of the Thorup-Zwick oracle is associated with the value in the list closest to $d_G(u, p_i)$. For each element of $\mathcal{L}$, a hash table is kept for the bunch vertices associated with that element. It can be shown that this way of linking the oracle of Thorup-Zwick and Mendel-Naor achieves the desired.

DOI 10.1007/978-3-642-27848-8_568-1

Applications

The practical need for efficient algorithms to answer the shortest path (distance) queries in graphs has increased significantly over the years, in large part due to emerging GPS navigation technology and other route planning software. Classical algorithms like Dijkstra do not scale well as they may need to explore the entire graph just to answer a single query. As road maps are typically of considerable size, obtaining compact distance oracles has received a great deal of attention from the research community.

Open Problems

A widely believed girth conjecture of Erdős [2] implies that an oracle with stretch $2k - 1$, size $O\left(n^{1+1/k}\right)$, and query time $O(1)$ would be optimal. Obtaining such an oracle (preferably one that is path reporting) is a main open problem in the area. Some progress has recently been made: Chechik [1] gives an oracle (not path reporting) with stretch $2k - 1$, size $O\left(k n^{1+1/k}\right)$, and $O(1)$ query time.

Recommended Reading

1. Chechik S (2014) Approximate distance oracles with constant query time. In: STOC, New York, pp 654–663
2. Erdős P (1964) Extremal problems in graph theory. In: Theory of graphs and its applications (Proceedings of the symposium on smolenice, 1963). Czechoslovak Academy of Sciences, Prague, pp 29–36
3. Mendel M, Naor A (2007) Ramsey partitions and proximity data structures. J Eur Math Soc 9(2):253–275. See also FOCS'06
4. Thorup M, Zwick U (2005) Approximate distance oracles. J Assoc Comput Mach 52:1–24
5. Wulff-Nilsen C (2013) Approximate distance oracles with improved query time. In: Proceedings of the 24th ACM-SIAM symposium on discrete algorithms (SODA), New Orleans, pp 202–208