New Algorithms for All Pairs Approximate Shortest Paths

Liam Roditty
Bar-Ilan University
Israel
liam.roditty@biu.ac.il

ABSTRACT

Let $G = (V, E)$ be an unweighted undirected graph with n vertices and m edges. Dor, Halperin, and Zwick [FOCS 1996, SICOMP 2000] presented an $\tilde{O}(\min\{n^{3/2}m^{1/2}, n^{7/3}\})$-time algorithm that computes estimated distances with an additive approximation of 2 without using Fast Matrix Multiplication (FMM). Recently, Deng, Kirkpatrick, Rong, V. Williams and Zhong [ICALP 2022] improved the running time for dense graphs to $\tilde{O}(n^{2.29})$-time, using FMM, where an exact solution can be computed with FMM in $\tilde{O}(n^{\omega})$ time ($\omega < 2.37286$) using Seidel's algorithm.

Since an additive 2 approximation is also a multiplicative 2 approximation, computing an additive 2 approximation is at least as hard as computing a multiplicative 2 approximation. Thus, computing a multiplicative 2 approximation might be an easier problem. Nevertheless, more than two decades after the paper of Dor, Halperin, and Zwick was first published, no faster algorithm for computing multiplicative 2 approximation in dense graphs is known, rather then simply computing an additive 2 approximation.

In this paper we present faster algorithms for computing a multiplicative 2 approximation without FMM. We show that in $\tilde{O}(\min\{n^{1/2}m, n^{9/4}\})$ time it is possible to compute a multiplicative 2 approximation. For distances at least 4 we can get an even faster algorithm that in $\tilde{O}(\min\{n^{7/4}m^{1/4}, n^{11/5}\})$ expected time computes a multiplicative 2 approximation.

Our algorithms are obtained by a combination of new ideas that take advantage of a careful new case analysis of the additive approximation algorithms of Dor, Halperin, and Zwick. More specifically, one of the main technical contributions we made is an analysis of the algorithm of Dor, Halperin, and Zwick that reveals certain cases in which their algorithm produces improved additive approximations without any modification. This analysis provides a full characterization of the instances for which it is harder to obtain an improved approximation. Using more ideas we can take care of some of these harder cases and to obtain an improved additive approximation also for them. Our new analysis, therefore, serves as a starting point for future research either on improved upper bounds or on conditional lower bounds.

CCS CONCEPTS

• **Theory of computation → Shortest paths**.

STOC '23, June 20–23, 2023, Orlando, FL, USA

ACM ISBN 978-1-4503-9913-5/23/06.
https://doi.org/10.1145/3564246.3585197

KEYWORDS

graph algorithms, shortest paths, all pairs approximate shortest paths

ACM Reference Format:
Liam Roditty. 2023. New Algorithms for All Pairs Approximate Shortest Paths. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC '23), June 20–23, 2023, Orlando, FL, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3564246.3585197

1 INTRODUCTION

Let $G = (V, E)$ be an unweighted undirected graph with n vertices and m edges. Let the distance $d_G(u, v)$, between every pair of vertices $u, v \in V$, be the length of the shortest path between u and v. Computing All Pairs of Shortest Paths (APSP) and their length is one of the most fundamental problems in Computer Science. The fastest known algorithms for APSP in unweighted graphs run in $\tilde{O}(\min\{mn, n^{\omega}\})$ time[1] [21], where $\omega < 2.373$ is the exponent of square matrix multiplication [5, 18, 23, 25].

In their seminal work, Aingworth, Chekuri, Indyk and Motwani [3] initiated the research on efficient APSP algorithms in unweighted undirected graphs, that settle for an approximated solution and do not use Fast Matrix Multiplication (FMM) algorithms, that hide large constants and are thus far from being practical. Let M be an $n \times n$ matrix. We say that M is an additive k approximation of the distances in G if $d_G(u, v) \leq M(u, v) \leq d_G(u, v) + k$, for every $u, v \in V$. We say that M is a multiplicative k approximation of the distances in G if $d_G(u, v) \leq M(u, v) \leq k d_G(u, v)$, for every $u, v \in V$. There is also a combination of additive and multiplicative approximations. We say that M is an (α, β)-approximation of the distances in G if $d_G(u, v) \leq M(u, v) \leq \alpha d_G(u, v) + \beta$. The following simple observation connects between a $(2, 0)$-approximation and a $(1, k)$-approximation.

Observation 1.1. *Let $u, v \in V$. If $d_G(u, v) \geq k$ and M is a $(1, k)$-approximation of the distances in G then M is a $(2, 0)$-approximation for the distance $d_G(u, v)$.*

Aingworth, Chekuri, Indyk and Motwani [3] presented an $\tilde{O}(n^{2.5})$ time algorithm that computes a matrix M with an additive 2 approximation for the distances in unweighted undirected graphs. Shortly after, in another seminal paper, Dor, Halperin and Zwick [14] presented an $\tilde{O}(\min\{n^{3/2}m^{1/2}, n^{7/3}\})$ time algorithm that computes a matrix M with an additive 2 approximation for the distances in unweighted undirected graphs. Recently, Deng, Kirkpatrick, Rong, V. Williams and Zhong [13] improved using FMM the running time for dense graphs from $\tilde{O}(n^{7/3})$ to $\tilde{O}(n^{2.29})$.

[1] $\tilde{O}$ notation hides polylogarithmic factors

More than two decades have elapsed since the publication of the paper of Dor, Halperin and Zwick [14] but the bound of

$$\tilde{O}(\min\{n^{3/2}m^{1/2}, n^{7/3}\})$$

on the running time (without FMM) was not improved yet. It is easy to see that if M is an additive 2 approximation it can be transformed into a multiplicative 2 approximation in $O(m)$ time, since all pairs at distance 1 can be found in $O(m)$ time and for $d_G(u,v) \geq 2$ it follows from Observation 1.1 with $k = 2$ that M is already a multiplicative 2 approximation. Nevertheless, even for the seemingly easier problem of computing a multiplicative 2 approximation no better algorithm is known.

Dor, Halperin and Zwick [14] presented also a conditional lower bound. They showed that any algorithm that computes a multiplicative or an additive approximation that is strictly better than 2 is capable to multiply two Boolean matrices in the same running time. Another upper bound presented by Dor, Halperin and Zwick [14] is an $\tilde{O}(n^2)$ time algorithm with a multiplicative 3 approximation. Baswana, Goyal and Sen [7] presented an $\tilde{O}(n^2)$ expected time algorithm with $(2,3)$-approximation and an $\tilde{O}(m^{2/3}n + n^2)$ expected time algorithm with $(2,1)$-approximation. For weighted undirected graphs, Baswana and Kavitha [8] and Berman and Kasiviswanathan [9] presented a $(2,W)$-approximation algorithm, where W is the largest edge weight, with running time of $\tilde{O}(n^2)$. For unweighted undirected graph this implies a $(2,1)$-approximation. Also, in the context of weighted undirected graphs, Baswana and Kavitha [8] presented a multiplicative 2 approximation algorithm with $\tilde{O}(n^2 + m\sqrt{n})$ expected running time.

The ultimate research goal is an $\tilde{O}(n^2)$-time algorithm that computes $(1,2)$-approximation. However, despite all the work an $\tilde{O}(n^2)$-time algorithm, even for the possibly easier problem of computing $(2,0)$-approximation, remains an elusive goal. It is completely plausible that it is impossible to compute in $\tilde{O}(n^2)$ time even $(2,0)$-approximation. In light of the lack of progress in obtaining improved algorithms without FMM for computing $(1,2)$-approximation in unweighted graphs we focus on the following question:

Question 1.2. *Can we compute* $(2,0)$*-approximation faster than computing* $(1,2)$*-approximation?*

In this paper we answer this question in the affirmative. We present an $\tilde{O}(\min\{n^{5/3}m^{1/3}, n^{9/4}\})$ time algorithm that computes (without FMM) a matrix M with a multiplicative 2 approximation for the distances in unweighted undirected graphs. This improves upon the current state of the art $O(\min\{n^{3/2}m^{1/2}, n^{7/3}\})$ time algorithm (without FMM) which is simply a byproduct of the additive 2 approximation algorithm [14]. If we use the algorithm of Baswana and Kavitha [8] we get an $\tilde{O}(\min\{n^{1/2}m, n^{9/4}\})$ time algorithm.

Dor, Halperin and Zwick [14] presented also the following tradeoffs between running time and accuracy for sparse and dense graphs. For every even $2 < k = O(\log n)$, there is an algorithm with a running time of $\tilde{O}(\min\{n^{2-\frac{2}{k+2}}m^{\frac{2}{k+2}}, n^{2+\frac{2}{3k-2}}\})$ that computes a $(1,k)$-approximation. As we explained above this is a $(2,0)$-approximation, for every $u,v \in V$ for which $d_G(u,v) \geq k$. This implies that for larger distances we can compute $(2,0)$-approximation faster. However, even in this case we are far from a quadratic running time. In order to understand what are the barriers that prevent us from obtaining the goal of $(2,0)$-approximation in $O(n^2)$ time we study the following restricted question:

Question 1.3. *Can we compute* $(2,0)$*-approximation faster than computing* $(1,k)$*-approximation for vertex pairs at distance at least* k*?*

In this paper we answer this question in the affirmative for sparse graphs. We reduce the running time for computing a $(2,0)$-approximation for vertex pairs at distance at least k from

$$\tilde{O}(\min\{n^{2-\frac{2}{k+2}}m^{\frac{2}{k+2}}, n^{2+\frac{2}{3k-2}}\})$$

time to

$$\tilde{O}(\min\{n^{2-\frac{2}{k+4}}m^{\frac{2}{k+4}}, n^{2+\frac{2}{3k-2}}\})$$

expected time. We obtain this result by showing that for every $u,v \in V$ for which $d_G(u,v) \leq k$ we can improve the additive approximation from $(1,k)$ to $(1,k-2)$ while keeping the running time of $\tilde{O}(n^{2-\frac{2}{k+2}}m^{\frac{2}{k+2}})$ unchanged.

A consequence of this new tradeoff is an algorithm that in $\tilde{O}(\min\{n^{11/5}, m^{1/4}n^{7/4}\})$ expected time computes a $(2,0)$ - approximation for every pair $u,v \in V$, for which $d_G(u,v) > 3$.

Our results are obtained by combining several new ideas with a better analysis of one of the algorithms of Dor, Halperin and Zwick [14] in certain cases. This analysis is of independent interest since it can help us in understanding the barriers for improving the running time of algorithms with an additive approximation.

Prior Work. Many different aspects of distance approximation in undirected graphs were studied since the work of Aingworth, Chekuri, Indyk and Motwani [3] more than two decades ago. Elkin [15] presented an $(1+\epsilon, W\beta(\epsilon,\rho,\varsigma))$-approximation $O(mn^\rho + n^{2+\varsigma})$ time algorithm for weighted graphs, where W is the ratio between the largest edge weight and the smaller edge weight[2]. Thorup and Zwick [24] studied the problem of approximating APSP also from the perspective of space. They show that for any integer $k \geq 1$ it is possible to preprocess a weighted undirected graph in $O(kmn^{1/k})$ expected time and to create an approximate distance oracle (ADO) of size $O(kn^{1+1/k})$. For every $u,v \in V$ a query returns in $O(k)$ time a multiplicative $2k-1$ approximation. Many of the subsequent works on approximating APSP were focused on improving the preprocessing time of the algorithm of Thorup and Zwick. (For more details see for example [8, 22, 26].) Baswana and Kavitha [8] showed that for $k > 2$ it is possible to compute ADO in $\tilde{O}(\min\{n^2, kmn^{1/k}\})$ expected running time. Other aspects of ADO were considered as well (see [10, 11, 16, 17, 19, 26]).

Baswana, Gaur, Sen, and Upadhyay [6] used the ADO of Thorup and Zwick [24] and improved the running time of Dor, Halperin and Zwick [14] for multiplicative 3 approximation by introducting also an additive approximation. They obtained an $\tilde{O}(m+n^{23/12})$ expected time algorithm that constructs an $O(n^{1.5})$-space data structure, that in $O(1)$ time reports a $(3,14)$-approximate distance.

Pătraşcu and Roditty [20] obtained a $(2,1)$-ADO for unweighted graphs with $O(n^{5/3})$ space and $O(1)$ query time. (See also [2] and [1]). Sommer [22] presented an $\tilde{O}(n^2)$ time algorithm that computes a $(2,1)$-ADO of size $\tilde{O}(n^{5/3})$. Roditty and Akav [4] presented

[2]Notice that for unweighted graph this result in incomparable to our result since the additive term of $\beta(\epsilon,\rho,\varsigma)$ is large for small distances.

an algorithm that for every $\epsilon \in (0, 1/2)$ computes in $\tilde{O}(m)+n^{2-\Omega(\epsilon)}$ time a $(2(1+\epsilon), 5)$-ADO of size $\tilde{O}(n^{1\frac{5}{6}})$. Chechik and Zhang [12] improved this result and obtained an $\tilde{O}(m + n^{1.987})$ time algorithm that computes a $(2, 3)$-ADO of size $\tilde{O}(n^{5/3})$.

2 OVERVIEW

Dor, Halperin and Zwick [14] proved among other things the following two theorems:

Theorem 2.1 ([14]). *There is an algorithm that computes an additive 2 approximation for the distances in unweighted undirected graphs in $\tilde{O}(\min\{n^{3/2}m^{1/2}, n^{7/3}\})$ time.*

Theorem 2.2 ([14]). *For every even $2 < k = O(\log n)$, there is an algorithm that computes an additive k approximation for the distances in unweighted undirected graphs in $\tilde{O}(\min\{n^{2-\frac{2}{k+2}} m^{\frac{2}{k+2}}, n^{2+\frac{2}{3k-2}}\})$ time.*

In order to answer Question 1.2 we use the following approach that stems from Observation 1.1. We treat differently vertex pairs at distance at least 4 and vertex pairs at distance strictly less than 4. We prove the following theorem for vertex pairs at distance at most 3.

Theorem 2.3. *There is an algorithm that computes for every $u, v \in V$, for which $d_G(u, v) \leq 3$, an additive 2 approximation in $\tilde{O}(\min\{m^{1/3}n^{5/3}, n^{9/4}\})$ time.*

Using Theorem 2.3 and Observation 1.1 together with Theorem 2.2 we prove the following corollary which provides a positive answer to Question 1.2.

Corollary 2.4. *There is an algorithm that computes a multiplicative 2 approximation for the distances in unweighted undirected graphs in $\tilde{O}(\min\{n^{1/2}m, n^{9/4}\})$ time.*

Proof. In $O(m)$ we can find all vertices at distance 1. Using Theorem 2.3 we obtain for every $u, v \in V$, for which $2 \leq d_G(u, v) \leq 3$, an additive approximation of 2 which is also a multiplicative approximation of 2 in $\tilde{O}(\min\{m^{1/3}n^{5/3}, n^{9/4}\})$ time. We can also use Baswana and Kavitha [8] result, which is faster than $\tilde{O}(m^{1/3}n^{5/3})$ since we only need a multiplicative 2 approximation. Consider now vertex pairs $u, v \in V$ for which $4 \leq d_G(u, v)$. We use Theorem 2.2 with $k = 4$ and obtain an additive 4 approximation in $\tilde{O}(n^{11/5})$ time. As follows from Observation 1.1 an additive 4 approximation is a multiplicative 2 approximation when $d_G(u, v) \geq 4$. □

Theorem 2.3 is proved by a careful cases analysis of Theorem 2.2 with $k = 4$. In this case analysis we characterize the cases in which the additive 4 approximation algorithm of [14] actually computes an additive 2 approximation for vertex pair at distance at most 3. For the rest of the cases that the additive approximation is not 2 we show how to efficiently compute an additive 2 approximation using other ideas.

Next, we extend the ideas that were used to prove Theorem 2.3 to every vertex pair at distance at most k, where $k \geq 6$ is even integer. We prove:

Theorem 2.5. *Let $k \geq 6$ be an even integer. There is an algorithm that computes an additive $k-2$ approximation in $\tilde{O}(n^{2-\frac{2}{k+2}} m^{\frac{2}{k+2}})$ expected time, for every $u, v \in V$, for which $d_G(u, v) \leq k$.*

Using Theorem 2.5 and Observation 1.1 together with Theorem 2.2 we prove the following corollary which provides a positive answer to Question 1.3.

Corollary 2.6. *There is an algorithm that computes a multiplicative 2 approximation in $\tilde{O}(\min\{n^{2-\frac{2}{k+4}} m^{\frac{2}{k+4}}, n^{2+\frac{2}{3k-2}}\})$ expected time, for every $u, v \in V$ for which $d_G(u, v) \geq k$, where $k \geq 4$ is even integer.*

Proof. If $n^{2-\frac{2}{k+4}} m^{\frac{2}{k+4}} < n^{2+\frac{2}{3k-2}}$ we use Theorem 2.5 with $k+2$ and compute in $\tilde{O}(n^{2-\frac{2}{k+4}} m^{\frac{2}{k+4}})$ expected time an additive k approximation which is also a multiplicative 2 approximation, for every $u, v \in V$, for which $k \leq d_G(u, v) \leq k+2$. We use the algorithm for sparse graphs from Theorem 2.2 with $k+2$ and compute in $\tilde{O}(n^{2-\frac{2}{k+4}} m^{\frac{2}{k+4}})$ time an additive $k+2$ approximation which is also a multiplicative 2 approximation, for every $u, v \in V$, for which $k+2 \leq d_G(u, v)$.

If $n^{2-\frac{2}{k+4}} m^{\frac{2}{k+4}} \geq n^{2+\frac{2}{3k-2}}$ we use the algorithm for dense graphs from Theorem 2.2 with k and compute in $\tilde{O}(n^{2+\frac{2}{3k-2}})$ time an additive k approximation which is also a multiplicative 2 approximation, for every $u, v \in V$, for which $k \leq d_G(u, v)$. □

Using Corollary 2.6 with $k = 4$ we can compute a multiplicative 2 approximation for all vertex pairs at distance greater than 3 in $\tilde{O}(\min\{n^{7/4}m^{1/4}, n^{11/5}\})$ expected time.

Paper Organization. The rest of this paper is organized as follows. In the next section we present some preliminaries. In Section 4 we prove Theorem 2.3. In Section 5 we prove Theorem 2.5.

3 PRELIMINARIES

Let $G = (V, E)$ be an unweighted undirected graph. Let $|V| = n$ and $|E| = m$. Let $G(V')$ be the graph induced in G by the set $V' \subset V$. Let $u, v \in V$ and let $d_G(u, v)$ be the length of the shortest path between u and v. For a vertex $v \in V$, let $N(v)$ be the neighbours of v including v itself. Let $N(u, t) = \{v \in V \mid d_G(u, v) \leq t\}$. Let $E(v)$ be the incident edges of v. We denote with $P(u, v)$ the vertices on a path between u and v. Given $P_1(u, w)$ and $P_2(w, v)$ we denote with $P_1(u, w) \cdot P_2(w, v)$ the concatenation of paths $P_1(u, w)$ and $P_2(w, v)$.

Let $S \subseteq V$. Let $u \in V \setminus S$. If $N(u) \cap S \neq \emptyset$ then we set $r(u, S)$ to be an arbitrary vertex from $N(u) \cap S$, otherwise, we set $r(u, S) = \emptyset$. Let $E(S) = \{(u, r(u, S)) \mid u \in V \wedge r(u, S) \neq \emptyset\}$. For every $s \in S$ let $q(s, S) = \{u \mid r(u, S) = s\}$.

Let D be the distance matrix of G and let M be the matrix that contains the approximated distances we compute using our algorithms. Throughout the paper we assume that M is always initialized such that $M(u, v) = 1$ if $(u, v) \in E$ and $M(u, v) = \infty$, otherwise. Let $S \subseteq V$. For every $u \in V$ let $W(u, S) = \{(u, s) \mid s \in S\}$. Each edge $(u, s) \in W(u, S)$ has a weight equal to the current value in $M(u, s)$. Notice that it is not required that the edge (u, s) will be in the edge set E.

Let $k \geq 2$ be an integer. Let $z_1 > z_2 > \cdots > z_{k-1} > z_k = 1$ and let $V_i = \{v \in V \mid deg(v) \geq z_i\}$, where $i \in [1, k]$. Notice that $V_1 \subseteq V_2 \subseteq \cdots \subseteq V_k = V$.

Let Z_k be V. Let $Z_i \subseteq V$, where $i \in [1, k-1]$, be a set that hits the neighbourhood of every vertex with degree at least z_i, that is,

Algorithm 1: apasp$_k$

Input: A graph $G = (V, E)$
Output: An $n \times n$ matrix M
for $i \leftarrow 1$ **to** k **do**
  foreach $u \in Z_i$ **do**
    Run Dijkstra for u in $H = (V, W(u, V) \cup E^* \cup F_i)$
    foreach $x \in V$ **do**
      $M(u, x) \leftarrow \min\big(M(u, x), d_H(u, x)\big)$

Algorithm 2: Additive-2 for distance ≤ 3

Input: A graph $G = (V, E)$
Output: An $n \times n$ matrix M
$M \leftarrow$ **apasp$_4$**(G)
`/* Step 1 - case` C_1 `*/`
foreach $x, y \in V$ **do** $M(x, y) \leftarrow \min_{s \in Z_1}\{M(x, s) + M(s, y)\}$
`/* Step 2 - case` C_2 `*/`
foreach $u \in V$ *and* $x \in V$ **do**
  $M(u, x) \leftarrow \min\{1 + M(r(u, Z_2), x), M(u, x)\}$
`/* Step 3 - case` $C_3(b)$ `*/`
foreach $u \in V \setminus V_2$ **do**
  Compute BFS from u in $G(V \setminus V_1)$ up to distance 2
  foreach $x \in V$ **do**
    $M(u, x) \leftarrow \min\{M(u, x), d_{G(V \setminus V_1)}(u, x)\}$
foreach $s \in Z_3$ **do**
  foreach $x \in V$ **do**
    $M(s, x) \leftarrow \min\{(\min_{u \in q(s, Z_3)} 1 + M(u, x)), M(s, x)\}$
  Run Dijkstra for s in $H_2 = (V, W(s, V)) \cup F_3)$
  foreach $x \in V$ **do** $M(s, x) \leftarrow \min\{M(s, x), d_{H_2}(s, x)\}$
foreach $u, v \in V$ **do**
  $M(u, v) \leftarrow \min\{(M(u, v), 1 + M(r(u, Z_3), v)\}$
return M

if $deg(u) \geq z_i$ then $Z_i \cap N(u) \neq \emptyset$. The set Z_i is formed either by adding every vertex of V, independently, with probability $\tilde{O}(1/z_i)$, or by using $N(u)$ of every $u \in V$, to compute deterministically the set Z_i in $\tilde{O}(n^2)$ time [3]. Unless stated otherwise we assume the sets are computed deterministically. Either way $|Z_i| = \tilde{O}(n/z_i)$.

Let $F_i = \{(u, v) \in E \mid deg(u) < z_{i-1} \vee deg(v) < z_{i-1}\}$, where $i \in [2, k]$ and let $F_1 = E$. Notice that $E = F_1 \supseteq F_2 \supseteq \cdots \supseteq F_k$. Let $E^* = \cup_{i=1}^{k} E(Z_i)$.

Theorem 2.2 of Dor, Halperin and Zwick is obtained by running Algorithm **apasp$_k$** (see Algorithm 1. We assume that the set Z_i are constructed as described above with $z_i = (\frac{m}{n})^{1-\frac{i}{k}})$. Let M_i be the matrix M after the i-th iteration ends. The following Lemma is proved in [14] as part of the correctness proof of **apasp$_k$**.

Lemma 3.1. *Let $u \in Z_i$ and $v \in V$, where $1 \leq i \leq k$. It holds that* $d_G(u, v) \leq M_i(u, v) \leq d_G(u, v) + 2(i - 1)$.

The next Lemma follows immediately from the implementation of **apasp$_k$**.

Lemma 3.2. *Algorithm **apasp$_k$** runs in $\tilde{O}(n^2 + \sum_{i=1}^{k} |F_i| \times |Z_i|)$.*

Theorem 2.2 follows from Lemma 3.1 and Lemma 3.2. More specifically, since $Z_k = V$ it follows from Lemma 3.1 that $d_G(u, v) \leq M_k(u, v) \leq d_G(u, v) + 2(k - 1)$, for every $u, v \in V$. Since $|Z_i| = \tilde{O}(n/z_i)$, for $i \in [1, k]$, $|F_i| \leq nz_{i-1}$, for $i \in [2, k]$ and $F_1 = E$ it follows from Lemma 3.2 that the running time is $\tilde{O}(n^{2-\frac{1}{k}} m^{\frac{1}{k}})$.

4 AN ADDITIVE 2 APPROXIMATION FOR DISTANCE AT MOST 3

In this section we prove Theorem 2.3.

Reminder of Theorem 2.3. *There is an algorithm that computes for every $u, v \in V$, for which $d_G(u, v) \leq 3$, an additive 2 approximation in $\tilde{O}(\min\{m^{1/3} n^{5/3}, n^{9/4}\})$ time.*

Let $u, v \in V$ and let $d_G(u, v) \leq 3$. Let $P(u, v)$ be a shortest path between u and v. We set $k = 4$ and run **apasp$_4$**. Thus, we have four degree thresholds z_1, z_2, z_3 and $z_4 = 1$ and their corresponding hitting sets Z_1, Z_2, Z_3 and $Z_4 = V$. We have four vertex sets $V_1 \subseteq V_2 \subseteq V_3 \subseteq V_4 = V$ and four edge sets $E = F_1 \supseteq F_2 \supseteq F_3 \supseteq F_4$. From Theorem 2.2 it follows that the resulted matrix M contains estimations with an additive error of 4. As we will elaborate below we post-process the matrix M and update M so that every vertex pair at distance at most 3 has an additive approximation of 2. In order to obtain an additive approximation of 2 for all vertex pairs at distance at most 3 we do a case analysis based on the following combinations of vertex degrees in $P(u, v)$ that cover all possible combinations of vertex degrees in $P(u, v)$.

(C_1) $P(u, v) \cap V_1 \neq \emptyset$
(C_2) $\neg C_1 \wedge \big(u \in V_2 \vee v \in V_2\big)$
(C_3) $\neg C_1 \wedge \big(u \in V \setminus V_2 \wedge v \in V \setminus V_2\big)$
  (a) $u \in V \setminus V_3 \wedge v \in V \setminus V_3$
  (b) $u \in V_3 \vee v \in V_3$

We now turn to present the post processing stage of our algorithm that starts after the call to **apasp$_4$**. The first step corresponds to C_1. We compute for every $s \in Z_1$ shortest paths to all vertices in G. For every $x, y \in V$ we update $M(x, y)$ with $\min_{s \in Z_1} d(x, s) + d(s, y)$.

The second step corresponds to C_2. We compute for every $s \in Z_2$ shortest paths to all vertices in (V, F_2). For every $u \in V_2$ and $x \in V$ we update $M(u, x)$ with $\min\{1 + d(r(u, Z_2), x), M(u, x)\}$.

For case $C_3(a)$ we show in the analysis that an additive 2 bound follows from the call to **apasp$_4$**.

The last step corresponds to $C_3(b)$. We first compute for every $u \in V \setminus V_2$ shortest paths in the graph $G(V \setminus V_1)$ up to depth 2 and update M accordingly. Next, we scan the vertices of Z_3 and for every $s \in Z_3$ and $x \in V$ we update $M(s, x)$ with $\min\{\min_{u \in q(s, Z_3)}(1 + M(u, x)), M(s, x)\}$. Recall that $q(s, Z_3) = \{u \mid r(u, Z_3) = s\}$. Finally, we compute for every $u \in V$ shortest paths in $(V, W(u, Z_3) \cup E(Z_3) \cup F_4)$. A pseudocode is presented in Algorithm 2

We now prove that M contains additive 2 approximations for distances 2 and 3.

Lemma 4.1. *Let $u, v \in V$. If $d_G(u, v) \leq 3$ then $M(u, v) \leq d_G(u, v) + 2$.*

Proof. Let $P(u,v)$ be a shortest path between u and v and let $d_G(u,v) \leq 3$. We prove the claim for the different vertex degrees combinations of $P(u,v)$.

(C_1) $P(u,v) \cap V_1 \neq \emptyset$. Let $w \in P(u,v)$ be a vertex of degree at least z_1. Let $w^* = r(w, Z_1)$. We have $d_G(u,w^*) \leq d_G(u,w) + 1$ and $d_G(v,w^*) \leq d_G(v,w) + 1$. It follows from Lemma 3.1 that the call to $\mathbf{apasp}_4$ set in $M(u,w^*)$ and $M(v,w^*)$ the values $d_G(u,w^*)$ and $d_G(v,w^*)$, respectively. Thus, after line 3 in step 1 it holds that $M(u,v) \leq d_G(u,w^*) + d_G(v,w^*) \leq d_G(u,w) + 1 + d_G(v,w) + 1 \leq d_G(u,v) + 2$.

(C_2) $\neg C_1 \wedge (u \in V_2 \vee v \in V_2)$. Assume that $u \in V_2$. Let $u^* = r(u, Z_2)$. We have $M(u^*,v) \leq d_{(V,W(u,V) \cup E^* \cup F_2)}(u^*,v)$ after the call to $\mathbf{apasp}_4$. Since $P(u,v) \cap V_1 = \emptyset$ and $(u,u^*) \in E^*$, we have from the triangle inequality that $d_{(V,W(u,V) \cup E^* \cup F_2)}(u^*,v) \leq 1 + d_G(u,v)$. Thus, after line 5 in step 2 we have $M(u,v) \leq d_G(u,u^*) + M(u^*,v) \leq d_G(u,v) + 2$.

(C_3) $\neg C_1 \wedge (u \in V \setminus V_2 \wedge v \in V \setminus V_2)$.

(a) $u \in V \setminus V_3 \wedge v \in V \setminus V_3$. Let $w \in P(u,v)$ be the vertex with the largest degree in $P(u,v)$. Since $P(u,v) \cap V_1 = \emptyset$ it follows that $w \in V_j \setminus V_{j-1}$, where $j \in \{2,3,4\}$. If $w \in V_4 \setminus V_3$ then all the vertices of $P(u,v)$ are in $V_4 \setminus V_3$ and $P(u,v) \subseteq F_4$. Therefore, when running Dijkstra in $\mathbf{apasp}_4$ for every vertex of $Z_4 = V$ in the graph $(V, W(u,V) \cup E^* \cup F_4)$, we compute for u the exact distance to v. Consider now the case that $w \notin V_4 \setminus V_3$. This implies that $w \neq u$, $w \neq v$ and $j \in \{2,3\}$. Let $w^* = r(w, Z_j)$. In the jth iteration of $\mathbf{apasp}_4$ we run Dijkstra for w^* in the graph $(V, W(u,V) \cup E^* \cup F_j)$. The edge $(w,w^*) \in E^*$. The paths $P(w,u)$ and $P(w,v)$ are in F_j. Thus, $M(w^*,u) \leq d_G(w,u) + 1$ and $M(w^*,v) \leq d_G(w,v)+1$ after this iteration. Assume w.l.o.g that $d_G(v,w) \leq d_G(w,u)$. Since $d_G(u,v) \leq 3$ it follows that $(v,w) \in E$. In the 4th iteration we run Dijkstra for u in the graph $(V, W(u,V) \cup E^* \cup F_4)$. The value of $M(u,w^*)$ is at most $d_G(w,u)+1$, thus the weight of the edge $(u,w^*) \in W(u,V)$ is at most $d_G(w,u) + 1$. The edge $(w^*,w) \in E^*$ and the edge $(w,v) \in F_4$, since $v \in V \setminus V_3$. Thus, after the 4th iteration $M(u,v) \leq d_G(u,v) + 2$.

(b) $u \in V_3 \vee v \in V_3$. Since $u \in V \setminus V_2$ we find in line 8 all vertices at distance at most 2 from u in $G(V \setminus V_1)$. If $d_G(u,v) = 2$ then since $P(u,v) \cap V_1 = \emptyset$ we have $d_{G(V \setminus V_1)}(u,v) = 2$, thus $M(u,v) = 2$. Thus, we can assume that $d_G(u,v) = 3$. Let $P(u,v) = \{u,a,b,v\}$. Again, since $P(u,v) \cap V_1 = \emptyset$ we have $d_{G(V \setminus V_1)}(u,b) = 2$, thus $M(u,b) = 2$ after line 9. (The same is true for v and a). Assume, w.l.o.g, that $u \in V_3$ and let $u^* = r(u, Z_3)$. After line 12 is executed since $u \in q(u^*, Z_3)$ it holds for the pair u^* and b that $M(u^*,b) \leq 3$. Next, we run Dijkstra for u^* in $H_2 = (V, W(u^*,V) \cup F_3)$. The edge (u^*,b) has weight $M(u^*,b) \leq 3$. Since $v \in V \setminus V_2$ we have $(b,v) \in F_3$. Thus, after line 14 we have $M(u^*,v) \leq 4$. Finally, we set $M(u,v)$ in line 15 to $\min\{M(u,v), 1 + M(r(u,Z_3),v)\}$ and we get that $M(u,v)$ is at most 5.

□

We now turn to analyze the running time of Algorithm 2.

Lemma 4.2. *Algorithm 2 runs in* $\tilde{O}(n^2 + n^2|Z_1| + n \cdot z_1 \cdot z_2 + \sum_{i=1}^4 |F_i| \cdot |Z_i|)$ *time.*

Proof. From Lemma 3.2 it follows that the cost of calling $\mathbf{apasp}_4$ is $\tilde{O}(n^2 + \sum_{i=1}^4 |F_i| \cdot |Z_i|)$. The first step of the post-processing takes $O(n^2 \times |Z_1|)$ time. The second step takes $O(n^2)$. In the third step we first run for every $u \in V \setminus V_2$ BFS in $G(V \setminus V_1)$ up to distance 2. Combining the fact that $deg(u) \leq z_2$ with the fact that the maximum degree of $G(V \setminus V_1)$ is at most z_1 it follows that the cost of finding all vertices at distance 2 from u is $z_2 \cdot z_1$. The total time this step takes is $O(n \cdot z_2 \cdot z_1)$.

We set $M(s,x)$ to $\min\{(\min_{u \in q(s,Z_3)} 1 + M(u,x)), M(s,x)\}$, for every $s \in Z_3$ and $x \in V$. Notice that to set $M(s,x)$ we scan $|q(s,Z_3)|$ entries of M. Thus, the total cost of this part is $O(\sum_{s \in Z_3} |q(s,Z_3)| \cdot n)$. As this cost equals to $O(n \sum_{s \in Z_3} |q(s,Z_3)|)$ and $\sum_{s \in Z_3} |q(s,Z_3)| = n$ we get that the cost of this part is $O(n^2)$.

We then run Dijkstra for every $s \in Z_3$ in $H_2 = (V, W(s,V)) \cup F_3)$ at a total cost of $O(|Z_3| \cdot |F_3|)$ time. The cost of lines 14 and 15 is $O(n^2)$. □

Using Lemma 4.2 we get the following running time:

Corollary 4.3. *Algorithm 2 runs in* $\tilde{O}(\min\{n^{9/4}, m^{1/3}n^{5/3}\})$ *time.*

Proof. The running time is $\tilde{O}(n^2 + n^2|Z_1| + n \cdot z_1 \cdot z_2 + \sum_{i=1}^4 |F_i| \cdot |Z_i|)$ as we showed in Lemma 4.2. First notice that we have $|F_1| = m$ and $|F_i| \leq nz_{i-1}$, for $i \in \{2,3,4\}$. We also have $|Z_i| = \tilde{O}(n/z_i)$, for $i \in \{1,2,3,4\}$.

If we set $z_1 = n^{3/4}$, $z_2 = n^{2/4}$, $z_3 = n^{1/4}$, and $z_1 = 1$ we get that the running time is $\tilde{O}(n^{9/4})$ since we have $n^2|Z_1| = n^2 n/n^{3/4} = n^{9/4}$, $n \cdot z_1 \cdot z_2 = n^{9/4}$, and $\sum_{i=1}^4 |F_i| \cdot |Z_i| = n^{9/4}$.

If we set $z_1 = n^{4/3}m^{-1/3}$, $z_2 = (m/n)^{2/3}$, $z_3 = (m/n)^{1/3}$, and $z_1 = 1$ we get that the running time is $\tilde{O}(m^{1/3}n^{5/3})$ since $n \cdot z_1 \cdot z_2 = nn^{4/3}m^{-1/3}(m/n)^{2/3} = \tilde{O}(m^{1/3}n^{5/3})$, $n^2|Z_1| = n^2 n^{-1/3} m^{1/3} = \tilde{O}(m^{1/3}n^{5/3})$, and $\sum_{i=1}^4 |F_i| \cdot |Z_i| = \tilde{O}(m^{1/3}n^{5/3})$. We check in advanced which of the two expression $n^{9/4}$ and $m^{1/3}n^{5/3}$ is smaller and run the faster algorithm. □

5 THE GENERAL CASE

In this Section we prove Theorem 2.5.

Reminder of Theorem 2.5. *Let* $k' \geq 6$ *be an even integer. There is an algorithm that computes an additive* $k' - 2$ *approximation in* $\tilde{O}(n^{2-\frac{2}{k'+2}} m^{\frac{2}{k'+2}})$ *expected time, for every* $u,v \in V$, *for which* $d_G(u,v) \leq k'$.

To easy the presentation for the rest of this section we set $k = (k'+2)/2$ and prove the following equivalent statement of Theorem 2.5.

Theorem 5.1. *For every integer* $3 < k = O(\log n)$, *there is an algorithm with a worst case running time of* $\tilde{O}(n^{2-\frac{1}{k}} m^{\frac{1}{k}})$ *and an expected running time of* $O(1)$ *that computes for every* $u,v \in V$, *for which* $d_G(u,v) \leq 2(k-1)$, *an additive* $2(k-2)$ *approximation.*

Let $z_i = (m/n)^{1-i/k}$, where $i \in [1,k]$. Thus, $(m/n)^{(k-1)/k} = z_1 > z_2 > \cdots > z_{k-1} = (m/n)^{1/k} > z_k = 1$. Recall that we have a vertex hierarchy $V_1 \subseteq V_2 \subseteq \cdots \subseteq V_k = V$, an edge hierarchy $E = F_1 \supseteq F_2 \supseteq \cdots \supseteq F_k$, and Z_i is a hitting set for neighbourhoods of size at least z_i. Let $p_i(u)$ be the closest vertex to u from the set Z_i. We define $\ell(v)$ as follows:

Algorithm 3: Additive-$(2(k-2))$ for distance $\leq 2(k-1)$

Input: A graph $G = (V, E)$
Output: An $n \times n$ matrix M

/* Step 1 - cases P_1, $P_2(i)$ and P_4 */
$M \leftarrow \mathbf{apasp}_k(G)$
/* Step 2 - cases $P_2(ii)$ and $P_2(iii)$ */
add to Z_{k-2} every vertex of V with probability $1/z_{k-2}$
foreach $u \in V$, *where* $\ell(u) = k-1$ **do**
  if $N(u,2) \geq z_{k-3}$ **then** compute set $X(u)$
  else
    if $|N(u,2) \cap Z_{k-2}| \leq 6z_{k-1} \ln n$ **then** $C(u) = N(u,2) \cap Z_{k-2}$
    else
      $M \leftarrow \mathbf{apasp}_{k-1}$
      return M
add to Z_{k-3} a hitting set of the sets $X(\cdot)$
foreach $u \in V$ **do** compute $p_{k-3}(u)$ and $d_G(u, p_{k-3}(u))$
foreach $u, v \in V$ **do**
  if $d_G(u, p_{k-3}(u)) \leq 2$ **then**
    $M(u,v) \leftarrow \min\{2 + M(p_{k-3}(u), v), M(u,v)\}$
  if $d_G(u, p_{k-3}(u)) > 2$ **then**
    $M(u,v) \leftarrow \min_{x \in C(u)}\{\min\{2 + M(x,v), M(u,v)\}\}$
/* Step 3 - cases P_3 */
for $i \leftarrow 1$ **to** $2(k-1)$ **do**
  foreach $\langle u, w\rangle \in V \setminus V_k \times V$ **do**
    foreach $x \in N(u)$ **do**
      $M(u,w) \leftarrow \min\{1 + M(x,w), M(u,w)\}$
return M

$$\ell(v) = \begin{cases} i, & \text{if } 1 < i \leq k \text{ and } v \in V_i \setminus V_{i-1}. \\ 1, & \text{otherwise.} \end{cases}$$

Notice that it follows from the definition of $\ell(v)$ that if $\ell(v) > 1$ then $z_{\ell(v)} \leq deg(v) < z_{\ell(v)-1}$ and if $\ell(v) = 1$ then $z_1 \leq deg(v)$. Let $r(x) = r(x, Z_{\ell(x)})$.

Our focus in this Section is on vertex pairs $u, w \in V$ that satisfy $d_G(u,w) \leq 2(k-1)$. Let $P(u,w) = \{u = u_1, u_2, \ldots, w_2, w_1 = w\}$ be a shortest path between u and w[3] and let $d_G(u,w) \leq 2(k-1)$. Our improved additive approximation is obtained by analysing different possible values of $\ell(u), \ell(w)$ and in one case even the possible values of $\ell(u_2)$ and $\ell(w_2)$. We consider the following cases:

- (P_1) $\ell(u) = \ell(w) = k$
- (P_2) $\ell(u) = \ell(w) = k-1$
  - (i) $\ell(u_2) = \ell(w_2) = k-1$
  - (ii) $\ell(u_2) = k-2 \vee \ell(w_2) = k-2$
  - (iii) $\ell(u_2) < k-2 \wedge \ell(w_2) < k-2$
- (P_3) $\Big(\ell(u) = k \wedge \ell(w) = k-1\Big) \vee \Big(\ell(u) = k-1 \wedge \ell(w) = k\Big)$
- (P_4) $\ell(u) < k-1 \vee \ell(w) < k-1$

It is easy to see that cases $P_1 - P_4$ cover all the possible cases. Our algorithm is composed of three steps, each step is responsible for a different set of cases. In the first step we run Algorithm $\mathbf{apasp}_k$ (Algorithm 1). This step is responsible for cases P_1, $P_2(i)$ and P_4. The matrix M computed in this step will be used as the starting point for handling the remaining cases as well.

In the second step in order to update the values of M and to improve the additive approximation for the remaining cases we preform several preceding computations and updates. We start by augmenting the set Z_{k-2} by adding every vertex of V to Z_{k-2} with probability $1/z_{k-2}$.

Then, let $u \in V$ and let $\ell(u) = k-1$. If $|N(u,2)| \geq z_{k-3}$ we compute a set $X(u) \subseteq N(u,2)$ of size $\Theta(z_{k-3})$. If $|N(u,2)| < z_{k-3}$ then if $|N(u,2) \cap Z_{k-2}| \leq 6z_{k-1} \ln n$ we compute the set $C(u) = N(u,2) \cap Z_{k-2}$. If $|N(u,2) \cap Z_{k-2}| > 6z_{k-1}$ we run $\mathbf{apasp}_{k-1}$.

The actual computation of $X(u)$ and $C(u)$ in the case that $|N(u,2) \cap Z_{k-2}| \leq 6z_{k-1} \ln n$ is done as follows. If there is $x \in N(u)$ with $\ell(x) \leq k-3$ then we are in the case that $|N(u,2)| \geq z_{k-3}$. Thus, we add to $X(u)$ the vertices u, x and an arbitrary set of size z_{k-3} from $N(x)$, and proceed to the next vertex in V. If there is no $x \in N(u)$ with $\ell(x) \leq k-3$ then $\ell(x) \geq k-2$. In such a case we run BFS from u up to distance 2 to compute the set $N(u,2)$. If $|N(u,2)| \geq z_{k-3}$ then we set $X(u)$ to $N(u,2)$. If $|N(u,2)| < z_{k-3}$ we start to compute $N(u,2) \cap Z_{k-2}$ by checking for each $x \in N(u,2)$ if $x \in Z_{k-2}$. If we exhaust the set $N(u,2)$ before we detect $6z_{k-1} \ln n + 1$ vertices in $N(u,2) \cap Z_{k-2}$ we set $C(u)$ to $N(u,2) \cap Z_{k-2}$. If we detect that there are at least $6z_{k-1} \ln n + 1$ vertices in $N(u,2) \cap Z_{k-2}$ we immediately abort and run $\mathbf{apasp}_{k-1}$.

Next, if we did not aborted in the computation of $N(u,2) \cap Z_{k-2}$, for every $u \in V$, we compute, deterministically, a set of size $\tilde{O}(n/z_{k-3})$ that hits all the sets $X(\cdot)$ that we have computed and add this hitting set to Z_{k-3}. We also compute for every $u \in V$ the vertex $p_{k-3}(u)$ and the value of $d_G(u, p_{k-3}(u))$, with respect to the augmented set Z_{k-3}.

In the last part of the second step we use all the new information we have computed in the previous parts of the second step to update M as follows. For every $u, v \in V$ we update $M(u,v)$ with $\min\{2 + M(p_{k-3}(u), v), M(u,v)\}$, if $d_G(u, p_{k-3}(u)) \leq 2$ or with $\min_{x \in C(u)}\{\min\{2 + M(x,v), M(u,v)\}\}$ if $d_G(u, p_{k-3}(u)) > 2$. The second step is responsible for cases $P_2(ii)$ and $P_2(iii)$.

In the third step for every vertex pair $\langle u, w\rangle \in V \setminus V_k \times V$ we scan $N(u)$ and for every $x \in N(u)$ we set $M(u,w)$ with $\min\{1 + M(x,w), M(u,w)\}$. We repeat $2(k-1)$ times on this part for every such vertex pair. This step is responsible for cases P_3. A pseudocode of our Algorithm is presented in Algorithm 3.

In the rest of the paper we analyse Algorithm 3. More specifically, we start with case P_4 and show in Corollary 5.2, which follows easily from Lemma 3.1, that Algorithm $\mathbf{apasp}_k$ already handles case P_4. We then turn to a relatively involve analysis in Section 5.1. We show that Algorithm $\mathbf{apasp}_k$ without any change also handles cases P_1 and $P_2(i)$.

We then proceed to cases $P_2(ii)$, $P_2(iii)$ and P_3 in Section 5.2. We prove that steps 2 and 3 handle these cases. We end in Section 5.3 with a formal proof of Theorem 2.5 that includes the running time analysis of Algorithm 3.

In the next Corollary which follows from Lemma 3.1 we show that Algorithm $\mathbf{apasp}_k$ handles case P_4:

[3] Notice that in case that $d_G(u,w) = 2$ then $u_2 = w_2$.

Corollary 5.2. *Let $u \in V$. If $\ell(u) \leq k-2$ then $M(u,v) \leq d_G(u,v) + 2(k-2)$, for every $v \in V$.*

Proof. Let $u^* = r(u)$. Since $u^* \in Z_{\ell(u)}$ and $\ell(u) \leq k-2$ it follows from Lemma 3.1 that $M(u^*,v) \leq d_G(u^*,v) + 2(k-3)$ before the kth iteration of $\mathbf{apasp}_k$. In the kth iteration we run Dijkstra from v in the graph $(V, W(v,V) \cup E^* \cup F_k)$. Since $(u,u^*) \in E^*$ and since $(v,u^*) \in W(v,V)$ with weight $M(u^*,v) \leq 1 + d_G(u,v) + 2(k-3)$ we get that $M(u,v) \leq d_G(u,v) + 2(k-2)$. □

5.1 Cases P_1 and $P_2(i)$

Let $Q(u,v)$ be a path between u and v. We define $\rho(u, Q(u,v))$ as follows. If there is an edge $(a,b) \in Q(u,v)$ such that $(a,b) \notin F_{\ell(u)}$ then we set $\rho(u, Q(u,v))$ to be the farthest vertex from u that is an endpoint of such an edge. If such an edge does not exist we set $\rho(u, Q(u,v))$ to v.

Recall that $P(u,w) = \{u = u_1, u_2, \dots, w_2, w_1 = w\}$ is a shortest path between u and w. In order to obtain our tighter analysis we define the set of blocking vertices $B(u,w) = \{x_0, x_1, x_2, x_3, \dots, x_t\}$ as follows. We set $x_0 = w$ and $x_1 = u$. For $i \geq 2$, we set $x_i = \rho(x_{i-1}, P(x_{i-1}, x_{i-2}))$ as long as $\ell(x_i) < \ell(x_{i-1})$. Next, we establish various properties and lemmas on the set $B(u,w)$.

5.1.1 Properties of $B(u,w)$. The following property follows immediately from the definition of $B(u,w)$.

Property 5.3. *$k \geq \ell(x_1) > \ell(x_2) > \cdots > \ell(x_t) \geq 1$ and $t \leq \ell(x_1)$.*

Using this property we can prove the following:

Lemma 5.4. *Let $\ell(x_1) = c$, where $c \leq k$. If $\ell(x_i) \geq c - i + 1$, for every $2 \leq i \leq t$ then $\ell(x_i) = c - i + 1$.*

Proof. We prove the claim by induction on i. From Property 5.3 we have $\ell(x_1) > \ell(x_2) > \cdots > \ell(x_t)$. For the base we prove the claim for $i = 2$. Since $\ell(x_1) = c$ and $\ell(x_1) > \ell(x_2) \geq c - 2 + 1$ we get that $c - 1 \leq \ell(x_2) < c$, thus $\ell(x_2) = c - 1$.

We assume the claim holds for every value smaller than i and prove it for i. From the induction hypothesis we have $\ell(x_{i-1}) = c - i + 2$. Since $\ell(x_{i-1}) > \ell(x_i) \geq c - i + 1$ we get that $c - i + 1 \leq \ell(x_i) < c - i + 2$, thus $\ell(x_i) = c - i + 1$. □

The next Property also follows immediately from the definition of $B(u,w)$.

Property 5.5. *For every $2 \leq i \leq t$ it holds that $x_i \in P(x_{i-1}, x_{i-2})$, which implies that $x_{i-1} \notin P(x_i, x_{i-2})$.*

We prove the following additional properties for $B(u,w)$.

Lemma 5.6. *Let $P(u,w) = \{u = u_1, u_2, \dots, w_2, w_1 = w\}$ be a shortest path between u and w of length at most $2(k-1)$. Let $B(u,w) = \{x_0, \dots, x_t\}$. Let $0 < i \leq t$ and let $r = i \mod 2$. We show that:*

(i) $P(x_r, x_i) = P(x_r, x_{r+2}) \cdot P(x_{r+2}, x_{r+4}) \cdots P(x_{i-2}, x_i)$

(ii) $P(x_r, x_i) \subseteq F_{\ell(x_{i-1})}$

(iii) $P(x_1, x_0) = P(x_{1-r}, x_{i-1}) \cdot P(x_{i-1}, x_i) \cdot P(x_i, x_r)$, *where $x_i \notin P(x_{1-r}, x_{i-1})$.*

Proof. We start with proving the following Claim:

Claim 5.7. *$\{x_{i-1}, x_{i+1}, \dots, x_t\} \cap P(x_i, x_{i-2}) = \emptyset$, for every $2 \leq i \leq t$.*

Proof. For $i = t$ we have from Property 5.5 that $\{x_{t-1}\} \cap P(x_t, x_{t-2}) = \emptyset$. Consider now the case that $2 \leq i \leq t-1$. From Property 5.5 we have $x_{i-1} \notin P(x_i, x_{i-2})$. From the definition of $B(u,w)$ it follows that $P(x_i, x_{i-2}) \subseteq F_{\ell(x_{i-1})}$.

Consider the set of vertices $\{x_{i+1}, \dots, x_t\}$ and let $b \in [i+1, t]$. From the definition of $B(u,w)$ it follows that x_b is an endpoint of an edge that is not in $F_{\ell(x_{b-1})}$ where $\ell(x_{b-1}) \leq \ell(x_i) < \ell(x_{i-1})$. Moreover, since $\ell(x_{b-1}) < \ell(x_{i-1})$ we have $F_{\ell(x_{i-1})} \subseteq F_{\ell(x_{b-1})}$. If a vertex from $\{x_{i+1}, \dots, x_t\}$ is in $P(x_i, x_{i-2})$ it means that we picked an endpoint of an edge in $F_{\ell(x_{i-1})}$ since $P(x_i, x_{i-2}) \subseteq F_{\ell(x_{i-1})}$. As $F_{\ell(x_{i-1})} \subseteq F_{\ell(x_{b-1})}$ we get that the edge picked is in $F_{\ell(x_{b-1})}$, a contradiction. □

Using Claim 5.7 we can prove the following Claim:

Claim 5.8. *$P(x_i, x_{i-2}) \cap B(u,w) = \{x_i, x_{i-2}\}$, for every $2 \leq i \leq t$.*

Proof. For $i = 2$ the claim follows from Claim 5.7. Thus, we can focus on $3 \leq i \leq t$. From Claim 5.7 it follows that $\{x_{i-1}, x_{i+1}, \dots, x_t\} \cap P(x_i, x_{i-2}) = \emptyset$. Thus, we need to show that $\{x_0, x_1, \dots, x_{i-3}\} \cap P(x_i, x_{i-2}) = \emptyset$.

Let $0 \leq j \leq i-3$ and assume, towards a contradiction, that there exists $x_j \in P(x_i, x_{i-2})$. Let j be the largest index for which it holds that $x_j \in P(x_i, x_{i-2})$. Assume first that $j = i-3$. From Property 5.5 we have $x_{i-1} \in P(x_{i-2}, x_{i-3})$. Since $x_{i-3} \in P(x_i, x_{i-2})$ we get that $x_{i-1} \in P(x_i, x_{i-2})$, a contradiction to Property 5.5 which states that $x_{i-1} \notin P(x_i, x_{i-2})$.

Assume now that $j = i-4$. From Property 5.5 we have $x_{i-2} \in P(x_{i-3}, x_{i-4})$. By our assumption $x_{i-4} \in P(x_i, x_{i-2})$. Thus, $x_i \notin P(x_{i-2}, x_{i-3})$. From Property 5.5 we have $x_{i-1} \in P(x_{i-2}, x_{i-3})$ and $x_i \in P(x_{i-1}, x_{i-2})$. This implies that $x_i \in P(x_{i-2}, x_{i-3})$, a contradiction.

We are left with the case that $j < i-4$. Notice that $j+2 \leq i-3$ thus, from the maximality of j it follows that either $x_i \in P(x_{j+2}, x_j)$ or $x_{i-2} \in P(x_{j+2}, x_j)$. From Claim 5.7 it follows that $\{x_{j+1}, x_{j+3}, \dots, x_t\} \cap P(x_{j+2}, x_j) = \emptyset$. Since $j \leq i-5$ we have that $j+5 \leq i$ and $j+3 \leq i-2$. Thus, if $x_i \in P(x_{j+2}, x_j)$ or $x_{i-2} \in P(x_{j+2}, x_j)$ we reach a contradiction since $\{x_{j+3}, \dots, x_t\} \cap P(x_{j+2}, x_j) \neq \emptyset$.

□

We can now prove the lemma. Consider the case that i is even (and $r = 0$). The odd case is analogous.

We first show that $P(x_0, x_i) = P(x_0, x_2) \cdot P(x_2, x_4) \cdots P(x_{i-2}, x_i)$. For $i = 2$ the claim holds trivially. We assume the claim holds for every even $j \leq i-2$ and prove it for i. From the induction hypothesis we have $P(x_0, x_{i-2}) = P(x_0, x_2) \cdot P(x_2, x_4) \cdots P(x_{i-4}, x_{i-2})$.

From Claim 5.8 we have $x_i \notin P(x_{j-2}, x_j)$, for every $j \in [2, i-2]$. Thus, we get that $P(x_0, x_i) = P(x_0, x_{i-2}) \cdot P(x_{i-2}, x_i)$, as required.

We now turn to show that $P(x_0, x_i) \subseteq F_{\ell(x_{i-1})}$. Recall that $x_j = \rho(x_{j-1}, P(x_{j-1}, x_{j-2}))$, for every even $2 \leq j \leq i$, thus $P(x_j, x_{j-2}) \subseteq F_{\ell(x_{j-1})}$. Now since $\ell(x_1) > \ell(x_3) > \cdots > \ell(x_{i-3}) > \ell(x_{i-1})$ and since $F_k \subseteq F_{k-1} \subseteq \cdots F_2 \subseteq F_1 = E$ we have $P(x_0, x_i) \subseteq F_{\ell(x_{i-1})}$.

Finally, we show that $P(x_1, x_0) = P(x_{1-r}, x_{i-1}) \cdot P(x_{i-1}, x_i) \cdot P(x_i, x_r)$. Recall that we assume that i is even, thus, $r = 0$. We show that $P(x_1, x_0) = P(x_1, x_{i-1}) \cdot P(x_{i-1}, x_i) \cdot P(x_i, x_0)$, the odd case is analogous. Using the first property for $i-1$, which is odd, we get $P(x_1, x_{i-1}) = P(x_1, x_3) \cdot P(x_3, x_5) \cdots P(x_{i-3}, x_{i-1})$. From Claim 5.8

it follows for every $j \in [3, i-1]$ that $P(x_{j-2}, x_j) \cap \{x_i\} = \emptyset$. Thus, $x_i \notin P(x_1, x_{i-1})$, which implies that $x_i \in P(x_{i-1}, x_0)$. We get that $P(x_1, x_0) = P(x_1, x_{i-1}) \cdot P(x_{i-1}, x_i) \cdot P(x_i, x_0)$, as required. □

Using Properties (i) and (iii) from Lemma 5.6 we can prove the following Corollary:

Corollary 5.9. *Let $0 < i \leq t$. We show that:* $d_G(x_{i-1}, x_i) + \sum_{j=1}^{i-1} d_G(x_{j+1}, x_{j-1}) = d(u, w)$.

Proof. We assume that i is even. (The case of an odd i is analogous). We have:

$$d_G(x_{i-1}, x_i) + \sum_{j=1}^{i-1} d_G(x_{j+1}, x_{j-1}) = d_G(x_1, x_3) + d_G(x_3, x_5) + \cdots + d_G(x_{i-3}, x_{i-1}) + d_G(x_{i-1}, x_i) + d_G(x_0, x_2) + d_G(x_2, x_4) + \cdots + d_G(x_{i-2}, x_i) = d_G(u, w),$$

where the last equality follows from combining Properties (i) and (iii) in Lemma 5.6, which gives

$$P(x_1, x_0) = P(x_1, x_3) \cdot P(x_3, x_5) \cdots P(x_{i-3}, x_{i-1}) \cdot P(x_{i-1}, x_i) \cdot P(x_i, x_{i-2}) \cdots P(x_4, x_2) \cdot P(x_2, x_0)$$

□

5.1.2 Bounds on M Using $B(u, w)$.

We now turn to prove several bounds on matrix M, the output of the call to Algorithm $\mathbf{apasp}_k$ (Algorithm 1), with respect to the vertices of $B(u, w)$. We start with the vertex $r(x_t)$.

Lemma 5.10. $M_{\ell(x_t)}(r(x_t), x_{t-1}) \leq 1 + d_G(x_{x_t}, x_{t-1})$.

Proof. Let $\ell(x_t) = a$. The graph in which we run Dijkstra's algorithm for $r(x_t)$ is $H = (V, W(r(x_t), V) \cup E^* \cup F_a)$. The edge $(r(x_t), x_t)$ is in E^*. Let $x_{t+1} = \rho(x_t, P(x_t, x_{t-1}))$. Since x_t is the last vertex in $B(u, w)$ it means that $\ell(x_{t+1}) \geq \ell(x_t) = a$. This implies that $P(x_t, x_{t-1}) \subseteq F_a$. Thus, $M_{\ell(x_t)}(r(x_t), x_{t-1}) \leq 1 + d_G(x_{x_t}, x_{t-1})$. □

Next, we consider vertex $r(x_{i-1})$, for every $2 \leq i \leq t$, and prove the following recursive inequality.

Lemma 5.11. *For every $2 \leq i \leq t$:*

$$M_{\ell(x_{i-1})}(r(x_{i-1}), x_{i-2}) \leq d_G(r(x_{i-1}), x_{i-1}) + M_{\ell(x_i)}(x_{i-1}, r(x_i)) + d_G(r(x_i), x_i) + d_G(x_i, x_{i-2}).$$

Proof. Let $\ell(x_{i-1}) = a$. When we run Dijkstra's algorithm for $r(x_{i-1}) \in Z_a$ we already ran Dijkstra's algorithm for every $u \in Z_b$ and $1 \leq b < a$. In particular $M_{\ell(x_i)}(r(x_{i-1}), r(x_i))$ is already updated since $\ell(x_i) < \ell(x_{i-1})$. The graph in which we run Dijkstra's algorithm for $r(x_{i-1})$ is $H = (V, W(r(x_{i-1}), V) \cup E^* \cup F_a)$. In H there is an edge between $r(x_{i-1})$ and $r(x_i)$. The weight of this edge is bounded by $M_{\ell(x_i)}(r(x_{i-1}), r(x_i)) \leq d_G(r(x_{i-1}), x_{i-1}) + M_{\ell(x_i)}(x_{i-1}, r(x_i))$. The edge $(r(x_i), x_i)$ is in E^*. Since $x_i = \rho(x_{i-1}, P(x_{i-1}, x_{i-2}))$ it follows that the path $P(x_i, x_{i-2})$ is in F_a. Thus, there is a path in H between $r(x_{i-1})$ and x_{i-2} of length at most $d_G(r(x_{i-1}), x_{i-1}) + M_{\ell(x_i)}(x_{i-1}, r(x_i)) + d_G(r(x_i), x_i) + d_G(x_i, x_{i-2})$, as required. □

Using Lemma 5.11 we can prove for $r(x_1)$, and x_0 the following explicit inequality:

Lemma 5.12. *Let* $D_i = \sum_{j=1}^{i-1} \Big(d_G(r(x_j), x_j) + d_G(r(x_{j+1}), x_{j+1}) + d_G(x_{j+1}, x_{j-1}) \Big)$, *for every $2 \leq i \leq t$.*
(i) $M_{\ell(x_1)}(r(x_1), x_0) \leq d_G(x_{i-1}, r(x_i)) + 2(\ell(x_i) - 1) + D_i$, *for every $2 \leq i \leq t$, and*
(ii) $M_{\ell(x_1)}(r(x_1), x_0) \leq 1 + d_G(x_{x_t}, x_{t-1}) + D_t$.

Proof. We prove by induction on i that:

$$M_{\ell(x_1)}(r(x_1), x_0) \leq M_{\ell(x_i)}(x_{i-1}, r(x_i)) + D_i$$

For the base case let $i = 2$. In this case we need to show that $M_{\ell(x_1)}(r(x_1), x_0) \leq M_{\ell(x_2)}(x_1, r(x_2)) + d_G(r(x_1), x_1) + d_G(r(x_2), x_2) + d_G(x_2, x_0)$. But this is exactly what we get from Lemma 5.11 with $i = 2$.

We assume the claim holds for every value smaller than i and prove the claim for i. From the induction hypothesis we have that:

$$M_{\ell(x_1)}(r(x_1), x_0) \leq M_{\ell(x_{i-1})}(x_{i-2}, r(x_{i-1})) + D_{i-1}.$$

From Lemma 5.11 we have:

$$M_{\ell(x_{i-1})}(r(x_{i-1}), x_{i-2}) \leq M_{\ell(x_i)}(x_{i-1}, r(x_i)) + d_G(r(x_{i-1}), x_{i-1}) + d_G(r(x_i), x_i) + d_G(x_i, x_{i-2}).$$

Thus, we get:

$$\begin{aligned} M_{\ell(x_1)}(r(x_1), x_0) &\leq M_{\ell(x_i)}(x_{i-1}, r(x_i)) + d_G(r(x_{i-1}), x_{i-1}) + \\ &\quad d_G(r(x_i), x_i) + d_G(x_i, x_{i-2}) + D_{i-1} \\ &= M_{\ell(x_i)}(x_{i-1}, r(x_i)) + D_i \end{aligned} \qquad (*)$$

Next, we bound $M_{\ell(x_i)}(x_{i-1}, r(x_i))$ with $d_G(x_{i-1}, r(x_i)) + 2(\ell(x_i) - 1)$, as follows from Lemma 3.1, and get the first part of the Lemma, $M_{\ell(x_1)}(r(x_1), x_0) \leq d_G(x_{i-1}, r(x_i)) + 2(\ell(x_i) - 1) + D_i$

To prove the second part, we set $i = t$ in $(*)$ thus $M_{\ell(x_1)}(r(x_1), x_0) \leq M_{\ell(x_t)}(x_{t-1}, r(x_t)) + D_t$.

We use Lemma 5.10 to bound $M_{\ell(x_t)}(x_{t-1}, r(x_t))$ with $1 + d_G(x_{x_t}, x_{t-1})$ and get:

$$M_{\ell(x_1)}(r(x_1), x_0) \leq 1 + d_G(x_{x_t}, x_{t-1}) + D_t.$$

□

5.1.3 New Bounds for $\mathbf{apasp}_k$ when $d(u, w) \leq 2(k-1)$.

In this section we use the properties and lemmas presented above to show that $\mathbf{apasp}_k$ computes an additive $2(k-2)$ approximation in cases P_1 and $P_2(i)$ when $d(u, w) \leq 2(k-1)$ without any change in the algorithm.

We first bound $M(u, w)$ in the case that $\ell(u) = \ell(w) = k$.

Lemma 5.13. *Let $\ell(u) = \ell(w) = k$ (case P_1).* $M(u, w) \leq d_G(u, w) + 2(k-2)$.

Proof. Let $X = B(u, w) = \{x_0, x_1, x_2, x_3, \ldots, x_t\}$, where $x_0 = w$ and $x_1 = u$. Let $Y = B(w, u) = \{y_0, y_1, y_2, y_3, \ldots, y_g\}$, where $y_0 = u$ and $y_1 = w$. Since $u = x_1$ we have $\ell(x_1) = k$, thus, $r(x_1) = x_1 = u$ and $M(u, w) = M_{\ell(x_1)}(r(x_1), x_0)$.

Consider the case that either there is a value $2 \le i \le t$ such that $\ell(x_i) \le k-i$ or there is a value $2 \le i' \le g$ such that $\ell(y_{i'}) \le k-i'$. Wlog, assume that $\ell(x_i) \le k-i$.

From Lemma 5.12(i) we have:

$$M_{\ell(x_1)}(r(x_1),x_0) \le d_G(x_{i-1},r(x_i)) + 2(\ell(x_i)-1)+$$
$$\sum_{j=1}^{i-1}\Big(d_G(r(x_j),x_j)+d_G(r(x_{j+1}),x_{j+1})+d_G(x_{j+1},x_{j-1})\Big).$$

Since for every $u \in V$ vertex $r(u)$ is either u or a neighbor of u we have $\sum_{j=1}^{i-1} d_G(r(x_j),x_j) \le i-1$ and $\sum_{j=1}^{i-1} d_G(r(x_{j+1}),x_{j+1}) \le i-1$. Since $r(x_1)=x_1$ we have $d_G(r(x_1),x_1)=0$, thus, $\sum_{j=1}^{i-1} d_G(r(x_j),x_j) \le i-2$. Using these bounds we get:

$$M_{\ell(x_1)}(r(x_1),x_0) \le 1+2(i-2)+d_G(x_{i-1},r(x_i))+$$
$$2(\ell(x_i)-1)+\sum_{j=1}^{i-1} d_G(x_{j+1},x_{j-1})$$

From the triangle inequality it follows that $d_G(x_{i-1},r(x_i)) \le 1+d_G(x_{i-1},x_i)$. Thus, we get:

$$M_{\ell(x_1)}(r(x_1),x_0) \le 2(i-1)+2(\ell(x_i)-1)+$$
$$d_G(x_{i-1},x_i)+\sum_{j=1}^{i-1} d_G(x_{j+1},x_{j-1})$$

From Corollary 5.9 we have $d_G(x_{i-1},x_i)+\sum_{j=1}^{i-1} d_G(x_{j+1},x_{j-1}) = d_G(u,w)$, thus,

$$M_{\ell(x_1)}(r(x_1),x_0) \le 2(i-1)+2(\ell(x_i)-1)+d_G(u,w).$$

Since $\ell(x_i) \le k-i$ and we get:

$$M_{\ell(x_1)}(r(x_1),x_0) \le 2(i-1)+2(k-i-1)+d_G(u,w) = d_G(u,w)+2(k-2).$$

Consider now the case that either $t<k$ or $g<k$ and assume, wlog, that $t<k$.

From Lemma 5.12(ii) we have:

$$M_{\ell(x_1)}(r(x_1),x_0) \le 1+d_G(x_{x_t},x_{t-1})+$$
$$\sum_{j=1}^{t-1}\Big(d_G(r(x_j),x_j)+d_G(r(x_{j+1}),x_{j+1})+d_G(x_{j+1},x_{j-1})\Big)$$

As before we have $\sum_{j=1}^{t-1} d_G(r(x_j),x_j) \le t-2$ and $\sum_{j=1}^{t-1} d_G(r(x_{j+1}),x_{j+1}) \le t-1$.

Thus, we get:

$$M_{\ell(x_1)}(r(x_1),x_0) \le 1+2(t-2)+1+d_G(x_{x_t},x_{t-1})+\sum_{j=1}^{t-1} d_G(x_{j+1},x_{j-1})$$

From Corollary 5.9 we have $d_G(x_{t-1},x_t)+\sum_{j=1}^{t-1} d_G(x_{j+1},x_{j-1}) = d_G(u,w)$, thus,

$$M_{\ell(x_1)}(r(x_1),x_0) \le 2(t-1)+d_G(u,w).$$

Finally, since $t \le k-1$ we get

$$M_{\ell(x_1)}(r(x_1),x_0) \le 2(k-2)+d_G(u,w).$$

By Property 5.3 we have $t \le \ell(x_1)=k$ and $g \le \ell(y_1)=k$, thus, we are left with the case that $t=k$, $g=k$, $\ell(x_i) \ge k-i+1$, for every $2 \le i \le t$ and $\ell(y_{i'}) \ge k-i'+1$, for every $2 \le i' \le g$. We will show that if $\ell(x_i) \ge k-i+1$, for every $2 \le i \le t$ and $\ell(y_{i'}) \ge k-i'+1$, for every $2 \le i' \le g$ then it is not possible that both $t=k$ and $g=k$. Assume, towards a contradiction, that $t=k$ and $g=k$. Recall that $x_0=y_1=w$ and $x_1=y_0=u$.

Since $\ell(x_1)=k$, $\ell(x_t) \ge k-t+1$, $\ell(x_{t-1}) \ge k-(t-1)+1$ and $t=k$ it follows from Lemma 5.4 that $\ell(x_t)=1$ and $\ell(x_{t-1})=2$.

Recall that by $B(u,w)$ definition $x_t=\rho(x_{t-1},P(x_{t-1},x_{t-2}))$ is the farthest vertex from x_{t-1} that is an endpoint of an edge (x'_t,x_t) of path $P(u,w)$ that satisfies $(x'_t,x_t) \notin F_{\ell(x_{t-1})}=F_2$, and $\ell(x'_t)=1$, as well.

From the definition of X and Y we have, $X \setminus \{u,w\} \subseteq P(u,w) \setminus (u,w)$ and $Y \setminus \{u,w\} \subseteq P(u,w) \setminus (u,w)$. Since $|X|-2=|Y|-2=k-1$ it cannot be that $(X \cap Y) \setminus \{u,w\} = \emptyset$ because in such a case $|(X \cup Y) \setminus \{u,w\}| = 2k-2$ but $|P(u,w) \setminus \{u,w\}| \le 2k-3$. Thus, $(X \cap Y) \setminus \{u,w\} \ne \emptyset$ and let $b' \in (X \cap Y) \setminus \{u,w\}$. From Lemma 5.4 it follows that $\ell(x_i)=\ell(y_i)=k-i+1$, for every $2 \le i \le k$. Thus, it must be that $b'=x_i=y_i$, where $2 \le i \le t$. If $b'=x_t=y_t$ and b' is the only vertex in the intersection then $|(X \cup Y) \setminus \{u,b',w\}| = 2k-4$. For every $a \in (X \cup Y) \setminus \{u,b',w\}$ we have $\ell(a) \ge 2$. This implies that both x_t and x'_t are not in $(X \cup Y) \setminus \{u,b',w\}$. Thus, there are only $2k-5$ vertices of P that can be in $(X \cup Y) \setminus \{u,b',w\}$, and there must me another vertex $b=x_i=y_i$, with $i<t$ in $(X \cap Y) \setminus \{u,w\}$.

Assume that $i<t$ is even. (The odd case is analogous.)

From Lemma 5.6(iii) we have $P(y_0,y_1)=P(y_0,y_i)\cdot P(y_i,y_{i-1})\cdot P(y_{i-1},y_1)$. From Property 5.5 we have $y_{i+1} \in P(y_i,y_{i-1})$. This means that y_{i+1} is an endpoint of an edge $(y'_{i+1},y_{i+1}) \in P(y_i,y_{i-1}) \subseteq P(x_i,x_0)$ and $(y'_{i+1},y_{i+1}) \notin F_{\ell(y_i)}$.

From Lemma 5.6(ii) we have that $P(x_i,x_0) \subseteq F_{\ell(x_{i-1})}$, and in particular, $(y'_{i+1},y_{i+1}) \in F_{\ell(x_{i-1})}$.

However, since $\ell(y_i)=\ell(x_i)<\ell(x_{i-1})$ we have $F_{\ell(x_{i-1})} \subseteq F_{\ell(y_i)}$ and we get that $(y'_{i+1},y_{i+1}) \in F_{\ell(x_{i-1})} \subseteq F_{\ell(y_i)}$, a contradiction.

□

We now turn to the case that $\ell(u)=\ell(u_2)=\ell(w)=\ell(w_2)=k-1$.

Lemma 5.14. *Let $\ell(u)=\ell(u_2)=\ell(w)=\ell(w_2)=k-1$ (case $P_2(i)$). Either $M(r(u),w) \le 1+d_G(u,w)+2(k-3)$ or $M(u,r(w)) \le 1+d_G(u,w)+2(k-3)$.*

Proof. Let $X=B(u,w)=\{x_0,x_1,x_2,x_3,\ldots,x_t\}$, where $x_0=w$ and $x_1=u$. Let $Y=B(w,u)=\{y_0,y_1,y_2,y_3,\ldots,y_g\}$, where $y_0=u$ and $y_1=w$.

Consider the case that either there is a value $2 \le i \le t$ such that $\ell(x_i) \le k-i-1$ or there is a value $2 \le i' \le g$ such that $\ell(y_{i'}) \le k-i'-1$. Wlog assume that $\ell(x_i) \le k-i-1$.

From Lemma 5.12(i) we have:

$$M_{\ell(x_1)}(r(x_1),x_0) \le d_G(x_{i-1},r(x_i)) + 2(\ell(x_i)-1)+$$
$$\sum_{j=1}^{i-1}\Big(d_G(r(x_j),x_j)+d_G(r(x_{j+1}),x_{j+1})+d_G(x_{j+1},x_{j-1})\Big).$$

As before, we have $\sum_{j=1}^{i-1} d_G(r(x_j), x_j) \leq i-1$ and $\sum_{j=1}^{i-1} d_G(r(x_{j+1}), x_{j+1}) \leq i-1$. Using these bounds we get:

$$M_{\ell(x_1)}(r(x_1), x_0) \leq 2(i-1) + d_G(x_{i-1}, r(x_i)) \\ +2(\ell(x_i)-1) + \sum_{j=1}^{i-1} d_G(x_{j+1}, x_{j-1})$$

From the triangle inequality it follows that $d_G(x_{i-1}, r(x_i)) \leq 1 + d_G(x_{i-1}, x_i)$. Thus, we get:

$$M_{\ell(x_1)}(r(x_1), x_0) \leq 2(i-1) + 2(\ell(x_i)-1) + d_G(x_{i-1}, x_i) + \\ 1 + \sum_{j=1}^{i-1} d_G(x_{j+1}, x_{j-1})$$

From Corollary 5.9 we have $d_G(x_{i-1}, x_i) + \sum_{j=1}^{i-1} d_G(x_{j+1}, x_{j-1}) = d_G(u, w)$, thus,

$$M_{\ell(x_1)}(r(x_1), x_0) \leq 2(i-1) + 1 + 2(\ell(x_i)-1) + d_G(u, w).$$

Since $\ell(x_i) \leq k-i-1$ and we get:

$$M_{\ell(x_1)}(r(x_1), x_0) \leq 2(i-1)+1+2(k-i-2)+d_G(u, w) = d_G(u, w) + \\ 2(k-3)+1.$$

Consider now the case that either $t < k-1$ or $g < k-1$ and assume, wlog, that $t < k-1$.

From Lemma 5.12(ii) we have:

$$M_{\ell(x_1)}(r(x_1), x_0) \leq 1 + d_G(x_{x_t}, x_{t-1}) + \\ \sum_{j=1}^{t-1} \Big(d_G(r(x_j), x_j) + d_G(r(x_{j+1}), x_{j+1}) + d_G(x_{j+1}, x_{j-1}) \Big)$$

As before we have $\sum_{j=1}^{t-1} d_G(r(x_j), x_j) \leq t-1$ and $\sum_{j=1}^{t-1} d_G(r(x_{j+1}), x_{j+1}) \leq t-1$.

Thus, we get:

$$M_{\ell(x_1)}(r(x_1), x_0) \leq 2(t-1)+1+d_G(x_{x_t}, x_{t-1}) + \sum_{j=1}^{t-1} d_G(x_{j+1}, x_{j-1})$$

From Corollary 5.9 we have $d_G(x_{t-1}, x_t) + \sum_{j=1}^{t-1} d_G(x_{j+1}, x_{j-1}) = d_G(u, w)$, thus,

$$M_{\ell(x_1)}(r(x_1), x_0) \leq 2(t-1) + 1 + d_G(u, w).$$

Finally, since $t \leq k-2$ we get

$$M_{\ell(x_1)}(r(x_1), x_0) \leq 2(k-3) + 1 + d_G(u, w).$$

By Property 5.3 we have $t \leq \ell(x_1) = k-1$ and $g \leq \ell(y_1) = k-1$, thus, we are left with the case that $t = k-1$, $g = k-1$, $\ell(x_i) \geq k-i$, for every $2 \leq i \leq t$ and $\ell(y_{i'}) \geq k-i'$, for every $2 \leq i' \leq g$.

We will show that if $\ell(x_i) \geq k-i$, for every $2 \leq i \leq t$ and $\ell(y_{i'}) \geq k-i'$, for every $2 \leq i' \leq g$ then it is not possible that both $t = k-1$ and $g = k-1$.

Assume, towards a contradiction, that $t = k-1$ and $g = k-1$. Recall that $x_0 = y_1 = w$ and $x_1 = y_0 = u$. Since $\ell(x_1) = k-1$, $\ell(x_t) \geq k-t$, $\ell(x_{t-1}) \geq k-(t-1)$ and $t = k-1$ it follows from Lemma 5.4 that $\ell(x_t) = 1$ and $\ell(x_{t-1}) = 2$.

Recall also that by $B(u, w)$ definition $x_t = \rho(x_{t-1}, P(x_{t-1}, x_{t-2}))$ is the farthest vertex from x_{t-1} that is an endpoint of an edge (x'_t, x_t) of path $P(u, w)$ that satisfies $(x'_t, x_t) \notin F_{\ell(x_{t-1})} = F_2$, and $\ell(x'_t) = 1$, as well.

Since $\ell(u_2) = \ell(w_2) = k-1$ it follows from the definition of X and Y that $X \setminus \{u, w\} \subseteq P(u, w) \setminus \{u, u_2, w, w_2\}$ and $Y \setminus \{u, w\} \subseteq P(u, w) \setminus \{u, u_2, w, w_2\}$. Since $|X|-2 = |Y|-2 = k-2$ it cannot be that $(X \cap Y) \setminus \{u, w\} = \emptyset$ because in such a case $|(X \cup Y) \setminus \{u, w\}| = 2k-4$ but $|P(u, w) \setminus \{u, u_2, w, w_2\}| \leq 2k-5$. Thus, $(X \cap Y) \setminus \{u, w\} \neq \emptyset$ and let $b' \in (X \cap Y) \setminus \{u, w\}$. From Lemma 5.4 it follows that $\ell(x_i) = \ell(y_i) = k-i$, for every $2 \leq i \leq k$. Thus, it must be that $b' = x_i = y_i$, where $2 \leq i \leq t$. If $b' = x_t = y_t$ and b' is the only vertex in the intersection then $|(X \cup Y) \setminus \{u, b', w\}| = 2k-5$. For every $a \in (X \cup Y) \setminus \{u, b', w\}$ we have $\ell(a) \geq 2$. This implies that both x_t and x'_t are not in $(X \cup Y) \setminus \{u, b', w\}$. Thus, there are only $2k-6$ vertices of P that can be in $(X \cup Y) \setminus \{u, b', w\}$, and there must me another vertex $b = x_i = y_i$, with $i < t$ in $(X \cap Y) \setminus \{u, w\}$.

Assume that $i < t$ is even. (The odd case is analogous.) From Lemma 5.6(iii) we have $P(y_0, y_1) = P(y_0, y_i) \cdot P(y_i, y_{i-1}) \cdot P(y_{i-1}, y_1)$. From Property 5.5 we have $y_{i+1} \in P(y_i, y_{i-1})$. This means that y_{i+1} is an endpoint of an edge $(y'_{i+1}, y_{i+1}) \in P(y_i, y_{i-1}) \subseteq P(x_i, x_0)$ and $(y'_{i+1}, y_{i+1}) \notin F_{\ell(y_i)}$.

From Lemma 5.6(ii) we have that $P(x_i, x_0) \subseteq F_{\ell(x_{i-1})}$, and in particular, $(y'_{i+1}, y_{i+1}) \in F_{\ell(x_{i-1})}$. However, since $\ell(y_i) = \ell(x_i) < \ell(x_{i-1})$ we have $F_{\ell(x_{i-1})} \subseteq F_{\ell(y_i)}$ and we get that $(y'_{i+1}, y_{i+1}) \in F_{\ell(x_{i-1})} \subseteq F_{\ell(y_i)}$, a contradiction.

□

Using Lemma 5.14 we prove:

Corollary 5.15. *Let $\ell(u) = \ell(u_2) = \ell(w) = \ell(w_2) = k-1$ (case $P_2(i)$). $M(u, w) \leq d_G(u, w) + 2(k-2)$.*

Proof. From Lemma 5.14 either $M(r(u), w) \leq 1 + d_G(u, w) + 2(k-3)$ or $M(u, r(w)) \leq 1 + d_G(u, w) + 2(k-3)$. Assume, w.l.o.g, that $M(r(u), w) \leq 1 + d_G(u, w) + 2(k-3)$. Since $\ell(u) = k-1$, this holds before the kth iteration. In the kth iteration of $\mathbf{apasp}_k$, we run Dijkstra from u in the graph $(V, W(u, V) \cup E^* \cup F_k)$. Since $(u, r(u)) \in E^*$ and since $(u, r(w)) \in W(u, V)$ with weight $M(r(u), w) \leq 1 + d_G(u, w) + 2(k-3)$ we get that $M(u, w) \leq d_G(u, w) + 2(k-2)$. □

5.2 Cases $P_2(ii)$, $P_2(iii)$ and P_3

We now turn our attention to the second and the third steps of Algorithm 3, right after the call to $\mathbf{apasp}_k$.

Recall that the sets Z_{k-2} and Z_{k-3} are augmented in the second step. In our analysis we refer to the augmented sets. Let A_u be the event that $|N(u, 2)| > z_{k-3}$ and let B_u be the event that $|N(u, 2) \cap Z_{k-2}| > 6z_{k-1} \ln n$.

Lemma 5.16. *If there is a vertex $u \in V$ for which $\neg A_u \cap B_u$ is true then $M(x, y) \leq d_G(x, y) + 2(k-2)$, for every $x, y \in V$.*

Proof. If there is a vertex $u \in V$ for which the event $\neg A_u \cap B_u$ is true we have both $|N(u, 2) \cap Z_{k-2}| > 6z_{k-1} \ln n$ and $|N(u, 2)| < z_{k-3}$. In such a case the algorithm detects u and calls to $\mathbf{apasp}_{k-1}$. The additive approximation is $2(k-2)$ for every $x, y \in V$ as follows from Theorem 2.2. □

We now analyse the case that event A_u is true.

Lemma 5.17. *Let $u, w \in V$ and let $d_G(u, w) \leq 2(k-1)$. If A_u is true then $M(u, w) \leq d_G(u, w) + 2(k-2)$.*

Proof. If A_u is true then $|N(u, 2)| > z_{k-3}$ and since Z_{k-3} is augmented with a hitting set of all sets $N(\cdot, 2)$ of size at least z_{k-3} we have $N(u, 2) \cap Z_{k-3} \neq \emptyset$. This implies that $d_G(p_{k-3}(u), u) \leq 2$. From Lemma 3.1 we have $M(p_{k-3}(u), w) \leq d_G(p_{k-3}(u), w) + 2(k-4)$. From the triangle inequality it follows that $d_G(p_{k-3}(u), w) \leq 2 + d_G(u, w)$. Thus, $M(p_{k-3}(u), w) \leq d_G(u, w) + 2(k-4) + 2$.

In line 14 of Algorithm 3 we set in this case the value of $M(u, w)$ to be $\min\{2 + M(p_{k-3}(u), w), M(u, w)\}$, and since $M(p_{k-3}(u), w) \leq d_G(u, w) + 2(k-4) + 2$ we get that $M(u, w) \leq d_G(u, w) + 2(k-4) + 4 \leq d_G(u, w) + 2(k-2)$. □

We now turn to the case that event $\neg B_u$ is true.

Lemma 5.18. *Let $u, w \in V$ and let $d_G(u, w) \leq 2(k-1)$. Let $P(u, w) = \{u = u_1, u_2, \ldots, w_2, w_1 = w\}$ and let $\ell(u_2) = k-2$. If $\neg B_u$ is true then $M(u, w) \leq d_G(u, w) + 2(k-2)$.*

Proof. If $\neg B_u$ is true then $|N(u, 2) \cap Z_{k-2}| \leq 6 z_{k-1} \ln n$. Let $u_2^* = r(u_2)$. Since $|N(u, 2) \cap Z_{k-2}| \leq 6 z_{k-1} \ln n$ we have $C(u) = N(u, 2) \cap Z_{k-2}$ and $u_2^* \in C(u)$. From Lemma 3.1 we have $M(u_2^*, w) \leq d_G(u_2^*, w) + 2(k-3)$. From the triangle inequality we have $d_G(u_2^*, w) \leq 1 + d_G(u_2, w)$. Thus, $M(u_2^*, w) \leq 1 + d_G(u_2, w) + 2(k-3)$.

Since u_2 is the second vertex on $P(u, w)$ we have $1 + d_G(u_2, w) = d_G(u, w)$, thus, $M(u_2^*, w) \leq 1 + d_G(u_2, w) + 2(k-3) = d(u, w) + 2(k-3)$.

In line 14 of Algorithm 3 we set in this case the value of $M(u, w)$ to $\min\{2 + M(u_2^*, w), M(u, w)\}$ and we get $M(u, w) \leq d_G(u, w) + 2(k-2)$. □

Using Lemma 5.17 and Lemma 5.18 we can now prove the bound for cases $P_2(ii)$ and $P_2(iii)$. We start with case $P_2(ii)$.

Lemma 5.19. *Let $u, w \in V$, where $d_G(u, w) \leq 2(k-1)$ and let $P(u, w) = \{u = u_1, u_2, \ldots, w_2, w_1 = w\}$. Let $\ell(u) = \ell(w) = k-1$ and $\ell(u_2) = k-2 \vee \ell(w_2) = k-2$ (case $P_2(ii)$). $M(u, w) \leq d_G(u, w) + 2(k-2)$.*

Proof. Assume w.l.o.g that $\ell(u_2) = k-2$. Recall that we assumed that $A_u \cup \neg B_u$ is true. If A_u is true then if follows from Lemma 5.17 that $M(u, w) \leq d_G(u, w) + 2(k-2)$. If $\neg B_u$ is true then since $\ell(u_2) = k-2$ it follows from Lemma 5.18 that $M(u, w) \leq d_G(u, w) + 2(k-2)$. □

We now turn to case $P_2(iii)$.

Lemma 5.20. *Let $u, w \in V$, where $d_G(u, w) \leq 2(k-1)$ and let $P(u, w) = \{u = u_1, u_2, \ldots, w_2, w_1 = w\}$. Let $\ell(u) = \ell(w) = k-1$, $\ell(u_2) < k-2$ and $\ell(w_2) < k-2$ (case $P_2(iii)$). It holds that $M(u, w) \leq d_G(u, w) + 2(k-2)$.*

Proof. Since $\ell(u_2) < k-2$ we have the $|N(u_2)| > z_{k-3}$, thus $\neg B_u$ is true. From Lemma 5.18 it follows that if $\neg B_u$ is true then $M(u, w) \leq d_G(u, w) + 2(k-2)$. □

Finally, we analyse case P_3.

Lemma 5.21. *Let $u, w \in V$, where $d_G(u, w) \leq 2(k-1)$ and let $P(u, w) = \{u = u_1, u_2, \ldots, w_2, w_1 = w\}$. Let $(\ell(u) = k \wedge \ell(w) = k-1) \vee (\ell(u) = k-1 \wedge \ell(w) = k)$ (case P_3). It holds that $M(u, w) \leq d_G(u, w) + 2(k-2)$.*

Proof. Let M_i be the matrix M after the ith iteration of the for loop in line 16 ends. We prove by induction on i that if $d_G(u, w) = i$ then $M_i(u, w) \leq d_G(u, w) + 2(k-2)$. If $i = 1$ then $(u, w) \in E$ and the exact distance is already set in $M(u, w)$. We assume the claim holds for all vertex pairs that are at distance less than i which satisfy case P_3 and prove the claim for i. If u_2 and w satisfy the conditions of case P_3 then since $u_2 \in N(u)$ in line 18 $M(u, w)$ is updated with $\min\left(1 + M_{i-1}(u_2, w), M_{i-1}(u, w)\right)$. From the induction hypothesis we have that $M_{i-1}(u_2, w) \leq d_G(u_2, w) + 2(k-2)$. Thus, $M_i(u, w) \leq 1 + M_{i-1}(u_2, w) \leq 1 + d_G(u_2, w) + 2(k-2) = d_G(u, w) + 2(k-2)$. If u_2 and w do not satisfy the conditions of case P_3 then at this stage we already have $M_{i-1}(u_2, w) \leq d_G(u_2, w) + 2(k-2)$ for all other possible cases. Thus, $M_i(u, w) \leq 1 + M_{i-1}(u_2, w) \leq 1 + d_G(u_2, w) + 2(k-2) = d_G(u, w) + 2(k-2)$. □

5.3 Proof of Theorem 2.5

We can now prove Theorem 2.5.

5.3.1 The Additive Approximation. Let $u, w \in V$, where $d_G(u, w) \leq 2(k-1)$. Cases $P_1 - P_4$ cover all possible value combinations of $\ell(u)$ and $\ell(w)$. If we are in case P_4 it follows from Corollary 5.2 that $M(u, w) \leq d_G(u, w) + 2(k-2)$. If we are in case P_1 it follows from Lemma 5.13 that $M(u, w) \leq d_G(u, w) + 2(k-2)$. If we are in case $P_2(i)$ it follows from Corollary 5.15 that $M(u, w) \leq d_G(u, w) + 2(k-2)$.

If there is a vertex $u \in V$ such that $\neg A_u \cap B_u$ is true it follows from Lemma 5.16 that $M(x, y) \leq d_G(x, y) + 2(k-2)$ for every $x, y \in V$. Thus, we can assume for the rest of the proof that $A_u \cup \neg B_u$ is true, for every $u \in V$.

If we are in case $P_2(ii)$ it follows from Lemma 5.19 that $M(u, w) \leq d_G(u, w) + 2(k-2)$. If we are in case $P_2(iii)$ it follows from Lemma 5.20 that $M(u, w) \leq d_G(u, w) + 2(k-2)$. If we are in case P_3 it follows from Lemma 5.21 that $M(u, w) \leq d_G(u, w) + 2(k-2)$.

5.3.2 The Running Time. We show that the running time is $\tilde{O}(m^{1/k} n^{2-1/k})$ in the worst case and $O(1)$ in expectation. From Lemma 3.2 it follows that the call to Algorithm $\mathbf{apasp}_k$ in step 1 takes $\tilde{O}(m^{1/k} n^{2-1/k})$ time.

In step 2 we first sample vertices to the set Z_{k-2} in $O(n)$ time. Computing for every $u \in V$, where $\ell(u) = k-1$, the set $X(u)$ if $|N(u, 2)| \geq z_{k-3}$, and the set $C(u)$ if $|N(u, 2)| < z_{k-3}$ and $|N(u, 2) \cap Z_{k-2}| < 6 z_{k-1} \ln n$ takes $O(n z_{k-2} z_{k-3})$ time from the following reasons. We scan for every vertex u the set $N(u)$. Since $\ell(u) = k-1$ we have $|N(u)| \leq z_{k-2}$ and the cost is $O(z_{k-2})$. If we find $x \in N(u)$ with $\ell(x) \leq k-3$ we add to $X(u)$ the vertices u, x and an arbitrary set of size z_{k-3} from $N(x)$. The cost for u in such a case is $O(z_{k-3})$. If there is no vertex $x \in N(u)$ with $\ell(x) \leq k-3$ then $\ell(x) \geq k-2$ and $deg(x) \leq z_{k-3}$, for every $x \in N(u)$. In such a case we run BFS from u up to distance of 2 to compute $N(u, 2)$ at a cost of $O(z_{k-2} z_{k-3})$. If $|N(u, 2)| \geq z_{k-3}$ then we set $X(u)$ to $N(u, 2)$. Now if $|N(u, 2)| < z_{k-3}$ then we need to compute $C(u)$ if possible. Since we have already computed the set $N(u, 2)$ we search $N(u, 2)$ for vertices of Z_{k-2} to compute $N(u, 2) \cap Z_{k-2}$ in $O(z_{k-2} z_{k-3})$ time. This gives a total cost of $O(n z_{k-2} z_{k-3})$. Since $z_{k-2} = (\frac{m}{n})^{2/k}$ and $z_{k-3} = (\frac{m}{n})^{3/k}$ we get a running time of $O(n \cdot (\frac{m}{n})^{5/k})$. For every $k > 3$ we have $O(n \cdot (\frac{m}{n})^{5/k}) \leq \tilde{O}(m^{1/k} n^{2-1/k})$. A deterministic computation of an hitting set of $X(\cdot)$ that we add to Z_{k-3} takes

$O(n^2)$ time. In case that we detect a vertex u such that $|N(u,2)| < z_{k-3}$ ($\neg B_u$) and $|N(u,2) \cap Z_{k-2}| > 6z_{k-1} \ln n$ (A_u) we run $\mathbf{apasp}_{k-1}$ at a cost of $\tilde{O}(m^{\frac{1}{k-1}} n^{2-\frac{1}{k-1}})$. We show that the event $\neg A_u \cap B_u$ is true for some vertex u with probability $1/n^3$.

Lemma 5.22. $\mathbb{P}(\bigcup_{u \in V}(\neg A_u \cap B_u)) < 1/n^3$.

Proof. We first bound the probability $\mathbb{P}(\neg A_u \cap B_u)$ for every $u \in V$. Notice that $\mathbb{P}(\neg A_u \cap B_u) = \mathbb{P}(B_u | \neg A_u) \cdot \mathbb{P}(\neg A_u) \leq \mathbb{P}(B_u | \neg A_u)$.

Each vertex of $N(u,2)$ is added to Z_{k-2} with probability $1/z_{k-2}$. Thus, if we let $X = |N(u,2) \cap Z_{k-2}|$ then $X \sim B(n', p')$, where $n' = |N(u,2)|$ and $p' = 1/z_{k-2}$. Event B_u is equivalent to the event $\{X > 6z_{k-1} \ln n\}$.

Thus, we need to bound $\mathbb{P}(\{X > 6z_{k-1} \ln n\} \mid \{|N(u,2)| < z_{k-3}\})$. Let $Y \sim B(z_{k-3} \ln n, 1/z_{k-2})$. Since $X \sim B(|N(u,2)|, 1/z_{k-2})$ and it is given that $|N(u,2)| < z_{k-3}$ we have $\mathbb{P}(\{X > 6z_{k-1} \ln n\} \mid \{|N(u,2)| < z_{k-3})\} \leq \mathbb{P}(\{Y > 6z_{k-1} \ln n\})$.

We now bound $\mathbb{P}(\{Y > 6z_{k-1} \ln n\})$. From Chebyshev's inequality we have $\mathbb{P}(\{Y > (1+\delta)\mu\}) < \left(\frac{\exp(\delta)}{(1+\delta)^{1+\delta}}\right)^{\mu}$, where $\mu = \mathbb{E}(Y) = \frac{z_{k-3}}{z_{k-2}} \ln n = z_{k-1} \ln n$ and $\delta = 5$. Thus, we get:

$$\mathbb{P}(Y > 6z_{k-1} \ln n) < \left(\frac{e^5}{6^6}\right)^{z_{k-1} \ln n} < e^{-4z_{k-1} \ln n} < e^{-4 \ln n} < 1/n^4.$$

Using the union bound we get that:

$$\mathbb{P}\left(\bigcup_{u \in V} \neg A_u \cap B_u\right) < 1/n^3.$$

□

From Lemma 5.22 it follows that with probability of at most $1/n^3$ we call to $\mathbf{apasp}_{k-1}$ that takes $O(m^{\frac{1}{k-1}} n^{2-\frac{1}{k-1}})$ time, thus we get $O(1)$ expected running time.

The computation of $p_{k-3}(u)$ and $d_G(u, p_{k-3}(u))$ for every $u \in V$ can be done easily in $\tilde{O}(m+n)$ time by running Dijkstra from a dummy vertex, connected to every vertex of Z_{k-3} a with zero weight edge.

In the last part of the second step we update for every pair $u, v \in V$ the value of $M(u,v)$ either with $\min\{2 + M(p_{k-3}(u), v), M(u,v)\}$ at a cost of $O(n^2)$ or with $\min_{x \in C(u)}\{\min\{2 + M(x,v), M(u,v)\}\}$ at a cost of $\tilde{O}(m^{1/k} n^{2-1/k})$, since $|C(u)| = \tilde{O}(z_{k-1}) = \tilde{O}((\frac{m}{n})^{1/k})$.

In the third step for every vertex pair $\langle u, w \rangle \in V \setminus V_k \times V$ we scan $N(u)$ and for every $x \in N(u)$ we set $M(u,w)$ with $\min\{1 + M(x,w), M(u,w)\}$. Since $u \in V \setminus V_k$ we have $|N(u)| \leq z_{k-1} = O((\frac{m}{n})^{1/k})$. Thus, this part takes $\tilde{O}(m^{1/k} n^{2-1/k})$ time. We repeat $2(k-1)$ times on this part for every such vertex pair. The total running time is, therefore, $\tilde{O}(m^{1/k} n^{2-1/k})$.

REFERENCES

[1] I. Abraham and C. Gavoille. 2011. On Approximate Distance Labels and Routing Schemes with Affine Stretch. In *DISC*. 404–415.

[2] Rachit Agarwal and Philip Brighten Godfrey. 2013. Brief announcement: a simple stretch 2 distance oracle. In *ACM Symposium on Principles of Distributed Computing, PODC '13, Montreal, QC, Canada, July 22-24, 2013*. 110–112.

[3] D. Aingworth, C. Chekuri, P. Indyk, and R. Motwani. 1999. Fast Estimation of Diameter and Shortest Paths (Without Matrix Multiplication). *SIAM J. Comput.* 28, 4 (1999), 1167–1181.

[4] Maor Akav and Liam Roditty. 2020. An almost 2-approximation for all-pairs of shortest paths in subquadratic time. In *SODA*. SIAM, 1–11.

[5] Josh Alman and Virginia Vassilevska Williams. 2021. A Refined Laser Method and Faster Matrix Multiplication. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, Dániel Marx (Ed.). SIAM, 522–539.

[6] Surender Baswana, Akshay Gaur, Sandeep Sen, and Jayant Upadhyay. 2008. Distance Oracles for Unweighted Graphs: Breaking the Quadratic Barrier with Constant Additive Error. In *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part I: Tack A: Algorithms, Automata, Complexity, and Games*. 609–621.

[7] S. Baswana, V. Goyal, and S. Sen. 2009. All-pairs nearly 2-approximate shortest paths in $O(n^2 polylogn)$ time. *Theor. Comput. Sci.* 410, 1 (2009), 84–93.

[8] S. Baswana and T. Kavitha. 2010. Faster Algorithms for All-pairs Approximate Shortest Paths in Undirected Graphs. *SIAM J. Comput.* 39, 7 (2010), 2865–2896.

[9] P. Berman and S. P. Kasiviswanathan. 2007. Faster Approximation of Distances in Graphs. In *Proc. WADS*. 541–552.

[10] Shiri Chechik. 2014. Approximate distance oracles with constant query time. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*. 654–663.

[11] Shiri Chechik. 2015. Approximate Distance Oracles with Improved Bounds. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*. 1–10.

[12] Shiri Chechik and Tianyi Zhang. 2022. Nearly 2-Approximate Distance Oracles in Subquadratic Time. In *SODA*. SIAM, 551–580.

[13] Mingyang Deng, Yael Kirkpatrick, Victor Rong, Virginia Vassilevska Williams, and Ziqian Zhong. 2022. New Additive Approximations for Shortest Paths and Cycles. In *ICALP (LIPIcs, Vol. 229)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 50:1–50:10.

[14] D. Dor, S. Halperin, and U. Zwick. 2000. All-Pairs Almost Shortest Paths. *SIAM J. Comput.* 29, 5 (2000), 1740–1759.

[15] M. Elkin. 2005. Computing almost shortest paths. *ACM Transactions on Algorithms* 1, 2 (2005), 283–323.

[16] Michael Elkin, Ofer Neiman, and Christian Wulff-Nilsen. 2016. Space-efficient path-reporting approximate distance oracles. *Theor. Comput. Sci.* 651 (2016), 1–10.

[17] Michael Elkin and Seth Pettie. 2016. A Linear-Size Logarithmic Stretch Path-Reporting Distance Oracle for General Graphs. *ACM Trans. Algorithms* 12, 4 (2016), 50:1–50:31.

[18] François Le Gall. 2014. Powers of tensors and fast matrix multiplication. In *International Symposium on Symbolic and Algebraic Computation, ISSAC '14, Kobe, Japan, July 23-25, 2014*. 296–303.

[19] Manor Mendel and Assaf Naor. 2006. Ramsey partitions and proximity data structures. In *FOCS*. 109–118.

[20] M. Pătraşcu and L. Roditty. 2010. Distance Oracles Beyond the Thorup–Zwick Bound. In *FOCS*.

[21] R. Seidel. 1995. On the All-Pairs-Shortest-Path Problem in Unweighted Undirected Graphs. *J. Comput. Syst. Sci.* 51, 3 (1995), 400–403.

[22] Christian Sommer. 2016. All-Pairs Approximate Shortest Paths and Distance Oracle Preprocessing. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 55)*. 55:1–55:13.

[23] A. Stothers. 2010. On the Complexity of Matrix Multiplication. *Ph.D. Thesis, U. Edinburgh* (2010).

[24] Mikkel Thorup and Uri Zwick. 2005. Approximate distance oracles. *J. ACM* (2005).

[25] Virginia Vassilevska Williams. 2012. Multiplying matrices faster than Coppersmith-Winograd. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*. ACM, 887–898.

[26] Christian Wulff-Nilsen. 2012. Approximate distance oracles with improved preprocessing time. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*. 202–208.

Received 2022-11-07; accepted 2023-02-06; revised 2022-11-07; accepted 2023-02-06