Beyond 2-Approximation for k-Center in Graphs

Ce Jin[*] Yael Kirkpatrick[†] Virginia Vassilevska Williams[‡] Nicole Wein[§]

Abstract

We consider the classical k-Center problem in undirected graphs. The problem is known to have a polynomial-time 2-approximation. There are even $(2+\varepsilon)$-approximation algorithms for every $\varepsilon > 0$ running in near-linear time. The conventional wisdom is that the problem is closed, as $(2-\varepsilon)$-approximation is NP-hard when k is part of the input, and for constant $k \geq 2$ it requires $n^{k-o(1)}$ time under the Strong Exponential Time Hypothesis (SETH).

Our first set of results show that one can beat the multiplicative factor of 2 in undirected unweighted graphs if one is willing to allow additional small additive error, obtaining $(2-\varepsilon, O(1))$ approximations. We provide several algorithms that achieve such approximations for all integers k with running time $O(n^{k-\delta})$ for $\delta > 0$. For instance, for every $k \geq 2$, we obtain an $O(mn + n^{k/2+1})$ time $\left(2 - \frac{1}{2k-1}, 1 - \frac{1}{2k-1}\right)$-approximation to k-Center, and for every $k \geq 10$ we obtain an $(3/2, 1/2)$-approximation algorithm running in $n^{k-1+1/(k+1)+o(1)}$ time. For 2-Center we also obtain an $\tilde{O}(mn^{\omega/3})$ time $(5/3, 2/3)$-approximation algorithm, where $\omega < 2.372$ is the fast matrix multiplication exponent. Notably, the running time of this 2-Center algorithm is faster than then the time needed to compute APSP.

Our second set of results are strong fine-grained lower bounds for k-Center. We show that our $(3/2, O(1))$-approximation algorithm is optimal, under SETH, as any $(3/2-\varepsilon, O(1))$-approximation algorithm requires $n^{k-o(1)}$ time. We also give a time/approximation trade-off: under SETH, for any integer $t \geq 1$, $n^{k/t^2-1-o(1)}$ time is needed for any $(2-1/(2t-1), O(1))$-approximation algorithm for k-Center. This explains why our $(2-\varepsilon, O(1))$ approximation algorithms have k appearing in the exponent of the running time. Our reductions also imply that, assuming ETH, the approximation ratio 2 of the known near-linear time algorithms cannot be improved by any algorithm whose running time is a polynomial independent of k, even if one allows for any constant additive error.

1 Introduction

The k-Center problem is a classical facility location problem in the clustering literature. Given a distance metric on n points, k-Center asks for a set of k of the points ("centers") that minimize the maximum over all points p of the distance from p to its nearest center; this is called the *radius*. An important special case of the problem is when the metric is the shortest paths distance metric of a given n-node, m-edge graph. This special case has a huge variety of applications as seen for instance in the many examples in the 1983 survey [TFL83]. When k is part of the input, the problem is well-known to be NP-hard as k-Dominating Set is just a special case: a set of k nodes is a dominating set in an unweighted graph if and only if they are a k-center with radius 1.

As noted by [HN79], this same reduction shows that even for unweighted undirected graphs it is NP-hard to obtain k centers whose radius is at most a factor of $(2-\varepsilon)$ from the optimum for $\varepsilon > 0$, as such an algorithm would be able to distinguish between radius 1 and greater than 1, and would thus be able to solve Dominating Set. As k-Dominating set is $W[2]$-complete, obtaining such a $(2-\varepsilon)$-approximation for k-Center is also $W[2]$-hard with parameter k, as noted by [Fel19a]. Via [PW10], the reduction also implies that under the Strong Exponential Time Hypothesis (SETH), a $(2-\varepsilon)$-approximation for k-Center requires $n^{k-o(1)}$ time for all integers $k \geq 2$. Further $(2-\varepsilon)$-approximation hardness results are known for special classes of graphs and restricted metrics (e.g. [Fel19a, FG88, KLP19]).

Several 2 and $(2+\varepsilon)$ (for all $\varepsilon > 0$) approximation algorithms for k-center have been developed over the years [Gon85, DF85, HS86, Tho04, ACLM23], even running in near-linear time in the graph size.

[*]MIT, cejin@mit.edu, supported by NSF grants CCF-2129139 and CCF-2127597, and a Simons Investigator Award. This work was initiated while the author was at the Simons Institute.

[†]MIT, yaelkirk@mit.edu, supported by NSF Grant No 2141064.

[‡]MIT, virgi@mit.edu, Supported by NSF Grant CCF-2330048, BSF Grant 2020356 and a Simons Investigator Award. This work was initiated while the author was at the Simons Institute.

[§]University of Michigan, nswein@umich.edu. This work was initiated while the author was at the Simons Institute.

Due to the aforementioned hardness of approximation results, it seems that the k-center approximation problem is closed: factor 2 is possible and factor $(2-\varepsilon)$ for $\varepsilon > 0$ is impossible under widely believed hypotheses.

However, what if we do not restrict ourselves to *multiplicative* approximations, and allow small *additive error* in addition? That is, we allow for the approximate radius $\tilde{R}$ to be between the true radius R and $\alpha R + \beta$ for $\alpha < 2$ and some very small β. Such *mixed*, (α, β) approximations are often studied in the shortest paths literature.

Hochbaum and Shmoys [HS86] actually considered mixed approximations and showed that it is easy to modify the reduction from k-Dominating set to show that it is still NP-hard to obtain (α, β)-approximations for any $\alpha < 2$ and any β, for k-Center on *weighted* graphs: From a dominating set instance $G = (V, E)$, create a weighted graph $G' = (V, V \times V, w)$ where if $(u, v) \in E$, $w(u, v) = W$ and if $(u, v) \notin E$, $w(u, v) = 2W$,[1]where $W = 2\beta/(2-\alpha)$; since $\alpha W + \beta < 2W$, any (α, β)-approximation to k-Center solves the k-Dominating set problem.

Besides NP-hardness, the modified reduction of [HS86] also shows that $n^{k-o(1)}$ time is needed for any $k \geq 2$ for $(2-\varepsilon, \beta)$-approximation to k-Center under SETH. However, the reduction crucially relies on the graph having weights. For $\varepsilon = 2 - \alpha$, these weights behave as $\Theta(\beta/\varepsilon)$ and are in fact quite large if ε is small. We thus ask:

Question 1: Are there $((2-\varepsilon), \beta)$-approximation algorithms for $\beta = O(1)$ and $\varepsilon > 0$ that run in $O(n^{k-\delta})$ time for some $\delta > 0$ in unweighted graphs?

In particular, what is the best mixed approximation that one can get in $O(n^{k-\delta})$ time for $\delta > 0$?

Further, the Hochbaum and Shmoys [HS86] reduction implies that for any fixed $\varepsilon > 0$, under SETH, any $O(n^{k-\delta})$ time $(2-\varepsilon, \beta)$-approximation algorithm for k-center must have additive error β at least a constant fraction of the maximum edge weight in the graph. Thus, it is also interesting whether one can obtain such $O(n^{k-\delta})$ time $(2-\varepsilon, \beta)$-approximation algorithms for integer-weighted graphs, where β is allowed to be proportional to the largest weight.

If the answer to question 1 is affirmative, then we ask:

Question 2: Can one get $((2-\varepsilon), \beta)$-approximation in near-linear time for $\varepsilon > 0$, $\beta = O(1)$ in unweighted undirected graphs?

The question is also interesting for weighted graphs when β is allowed to be proportional to the largest weight in the graph.

A last question is:

Question 3: When can one avoid computing all pairwise distances (APSP) in the graph?

The best algorithms for APSP in m-edge, n-vertex graphs run in $\tilde{\Theta}(mn)$ time even when the graphs are unweighted and undirected, when $m \leq n^{\omega-1}$. This was (conditionally) explained by [LVW18]. The near-linear time $(2+\varepsilon)$-approximation k-Center algorithms (e.g. [Tho04, ACLM23]) show that APSP computation is not needed for any k if one is happy with a $(2+\varepsilon)$-approximation. For the case of $k = 1$, 1-Center is also known as the Radius problem in graphs. The hardness for $(2-\varepsilon)$-approximation does not apply for $k = 1$ (as k-dominating set hardness under SETH for instance only applies for $k \geq 2$). Consequently, there has been a lot of work on $O(mn^{1-\delta})$ time $(2-\varepsilon)$-approximation algorithms for 1-Center that avoid the computation of all-pairs shortest paths, and have truly subquadratic running times in sparse graphs [ACIM99, AGV23, CGR16, BRS$^+$21, AVW16, CLR$^+$14, RV13]. For what other values of $k \geq 2$ can one obtain such algorithms that achieve $((2-\varepsilon), \beta)$-approximation?

1.1 Our results

We present the first fast $(2-\varepsilon, \beta)$-approximation algorithms for $\varepsilon > 0$ and $\beta = O(1)$ for k-Center in unweighted undirected graphs, answering Question 1 in the affirmative and addressing Question 3. We complement our algorithms with a variety of fine-grained lower bounds, showing strong hardness results, in particular providing convincing evidence that the answer to question 2 is a resounding NO.

Lower bounds. Our conditional hardness results are largely based on the popular Strong Exponential Time Hypothesis (SETH) of [IP01, CIP10] that states that no $O((2-\varepsilon)^n)$ time algorithm for constant $\varepsilon > 0$ can solve k-SAT on n variables for arbitrary $k \geq 3$. Due to a result by Williams [Wil05], SETH is known to imply strong hardness for the k-Orthogonal Vector (k-OV) problem of Fine-Grained Complexity (see also [Vas18]), and some

[1]It is actually not necessary to add edges (u, v) of weight $2W$ when $(u, v) \notin E$. It suffices to keep the same graph but add weights W to all edges.

k	**Approximation**	**Runtime**	**Comments**	**Reference**
2	$(5/3, 2/3)$	$mn^{\omega/3}$		Theorem 4.1
3	$(7/4, M)$	$n^{\frac{(5\omega-\omega^2+1)-\mu\omega(\omega-2)}{(3\omega-\omega^2+1)}}$	Integer edge weights $\leq M$.	Theorem 4.26
3	$(3/2, 1/2)$	$n^{\omega+\frac{1}{\omega}}$		Theorem 4.5
$k \geq 4$	$(3/2, 1/2)$	$n^{k-(3-\omega)+\frac{1}{k+1}}$		Theorem 4.5
$k \geq 10$	$(3/2, 1/2)$	$n^{k-1+\frac{1}{k+1}+o(1)}$		Theorem 4.5
k	$(2-\frac{1}{2k-1}, 1-\frac{1}{2k-1})$	$mn + n^{k/2+1}$	Combinatorial.	Theorem 4.7
$k \geq 3$	$(2-\frac{1}{2k-1}, 1-\frac{1}{2k-1})$	$n^{k/2+22/9(k+1)}$	Assuming $\omega = 2$.	Theorem 4.12
$k \geq 3$	$(2-\frac{1}{2k-1}, 1-\frac{1}{2k-1})$	$n^{k/2+\beta_0/(k+1)+(\omega-2)}$	$\beta_0 = \frac{-8\omega^2+18\omega+18}{3\omega+3} \approx 1.5516$.	Theorem 4.18
$k \geq 12$	$(2-\frac{1}{2k-1}, 1-\frac{1}{2k-1})$	$n^{k/2+\beta_1/(k+1)+o(1)}$	$\beta_1 = \frac{34\omega^2-24\omega-66}{3\omega+3} \approx 6.7533$.	Theorem 4.18
k	$(2-\frac{1}{2\ell}, 1-\frac{1}{2\ell})$	$mn + n^{k-\ell+\frac{\ell(\ell+1)}{2(k+1)}+1}$	Combinatorial, $\ell \leq k$.	Theorem 4.21

Table 1: Algorithmic Results.
We use "combinatorial algorithms" to refer to algorithms that do not use Fast Matrix Multiplication.

of our lower bounds are from k-OV. Some of our lower bounds are from an approximation version of k-OV, Gap Set Cover, whose hardness due to [KLM19, Lin19] is also based on SETH. One of our lower bounds is based on the Exponential Time Hypothesis (ETH) that states that 3SAT on n variables cannot be solved in $2^{o(n)}$ time. ETH is implied by SETH and is an even more plausible hardness hypothesis. All our conditional lower bounds assume the word-RAM model with $O(\log n)$ bit words.

We start with a simple conditional lower bound (see Theorem 3.6) from k-OV (and hence SETH) that implies that for every $k \geq 2$, $n^{k-o(1)}$ time is needed for any $(3/2-\delta, \beta)$-approximation algorithm for k-Center for $\varepsilon, \delta > 0$ and any constant β.

This result implies that to get $O(n^{k-\delta})$ time, one can at best hope for a $(3/2, O(1))$ approximation. In fact, since our reduction constructs a sparse graph, we get that this is also true for sparse graphs, and a $(3/2-\delta, \beta)$-approximation in fact requires $m^{k-o(1)}$ time.

Next, we prove a more general result which is our main hardness result:

Theorem 1.1. *There is a function $f\colon \mathbb{N}^+ \to \mathbb{N}^+$ such that the following holds. For all integers $t, \ell \geq 1$ and all $k \geq 2f(t)$, under SETH $n^{\frac{k}{f(t)}-1-o(1)}$ time is necessary to distinguish between radius $\leq (2t+1)\ell$ and radius $\geq (4t+1)\ell$ for k-center, even on graphs with $m = O(n)$ edges. Here, $f(t) = t$ for $t \in \{1,2,3,4\}$, and $f(t) \leq t^2$ for all $t \in \mathbb{N}^+$.*

The proof of our theorem, to our knowledge, is the first use of Gap Set Cover for approximation hardness results for graph problems. Some consequences of our result are as follows:

1. Under SETH, there is no $(2-\frac{1}{2t+1}-\varepsilon, \beta)$-approximation algorithm for k-center for any $\varepsilon > 0$, $\beta = O(1)$, running in $O(n^{k/t^2-1-\delta})$ time. E.g., a $(5/3-\varepsilon, \beta)$ approximation requires $O(n^{k-1-o(1)})$ time.
2. Because of the ETH-based hardness of Gap Set Cover [KLM19, Lin19], we also get that under ETH there can be no $f(k)n^{o(k)}$ time $(2-\varepsilon, \beta)$-approximation algorithm for k-Center for $\varepsilon > 0, \beta = O(1)$.

Recall that there are $\tilde{O}(mk)$ time[2] 2-approximation algorithms [Gon85, DF85, HS86], and $\tilde{O}(m)$ time $(2+\varepsilon)$-approximation algorithms [Tho04, ACLM23], for k-center. Point 2 above shows that these are *optimal* in a strong sense: even if we allow arbitrary polynomial time $O(n^c)$ and allow additional additive error β, one cannot beat the multiplicative factor of 2 for all k simultaneously. We thus answer Question 2 strongly in the negative.

Algorithms. We present the first ever improvements over the known multiplicative factor of 2 for k-center, with very small additive error, answering Question 1 in the affirmative. Our algorithmic results are summarized in Table 1.

[2]The notation $\tilde{O}$ hides polylogarithmic factors.

Our first algorithmic result is that for every $k \geq 3$, there is an $O(n^{k-\delta})$ time algorithm for $\delta > 0$ that achieves a $(3/2, 1/2)$-approximation. The algorithm utilizes fast matrix multiplication, and the bound is in terms of $\omega < 2.372$, the exponent of square matrix multiplication [VXXZ24]. We complement the algorithm with a conditional lower bound showing that the approximation ratio of the algorithm is tight.

Theorem 1.2. *Given an unweighted, undirected graph G, there is a randomized algorithm that computes a $(3/2, 1/2)$-approximation to the k-center w.h.p. and runs in time*

- $\tilde{O}(n^{\omega+1/(\omega+1)})$ *time for* $k = 3$,
- $\tilde{O}(n^{k-(3-\omega)+1/(k+1)})$ *for* $k \geq 4$, *and*
- $n^{k-1+1/(k+1)+o(1)}$ *time for* $k \geq 10$.

This result appears in the paper as Theorem 4.5.

Due to Theorem 3.6, under SETH, the multiplicative part of the approximation guarantee is **tight** for sub-n^k time algorithms, as any $(3/2 - \varepsilon, O(1))$-approximation algorithm requires $n^{k-o(1)}$ time under SETH.

We then turn to address Question 3, and in particular the follow-up question asking for what values of $k \geq 2$ one can achieve $(2 - \varepsilon, O(1))$-approximation algorithms that run faster than computing APSP. In particular, for sparse graphs (when $m \leq O(n)$), when can such algorithms run in $O(n^{2-\delta})$ time for $\delta > 0$?

The algorithms of Theorem 1.2 don't address this question. In particular, they explicitly compute APSP in unweighted graphs using Seidel's $O(n^\omega)$ time algorithm. One could improve the running time by instead using an approximate APSP algorithm such as the additive approximations of [DHZ00, SY24] or the mixed approximations of [Elk05], with a slight cost to the approximation. However, even then, since all n^2 distances are computed, the algorithm would always run in $\Omega(n^2)$ time.

We show that for 2-Center, one can in fact obtain an algorithm that is polynomially faster than mn and hence polynomially faster than n^2 in sparse graphs. The following result appears in the paper as Theorem 4.1.

Theorem 1.3. *There exists a randomized algorithm running in $\tilde{O}(mn^{\omega/3})$ that computes a $(5/3, 2/3)$-approximation to the 2-center of any undirected, unweighted graph, w.h.p. If the 2-center radius is divisible by 3, the algorithm gives a true multiplicative 5/3-approximation.*

Thus, just like with 1-center [ACIM99, AGV23, CGR16, BRS+21, AVW16, CLR+14, RV13], one can get a better-than-2 approximation algorithm for 2-center, faster than n^2 time in sparse graphs. We also note that the algorithm can be adapted to give a $(5/3, M)$-approximation for the 2-center of a graph with positive integer weights bounded by M. As noted earlier, due to the reduction of Hochbaum and Shmoys [HS86], $\Omega(M)$ additive error is necessary for any mixed approximation algorithm with multiplicative stretch $2 - \varepsilon$.

We then extend the techniques used to construct the algorithm for 2-center to obtain a general approximation scheme that works for any k-center. The following result appears in the paper as Theorem 4.7 and Theorem 4.12. The value α below is the largest real number so that an $n \times n^\alpha$ matrix can be multiplied by an $n^\alpha \times n$ matrix in $O(n^{2+\varepsilon})$ time for all $\varepsilon > 0$.

Theorem 1.4. *For any $k \geq 2$, there is a randomized combinatorial algorithm that in $\tilde{O}(mn + n^{k/2+1})$ time, computes w.h.p. a $\left(2 - \frac{1}{2k-1}, 1 - \frac{1}{2k-1}\right)$-approximation to k-center for any given unweighted, undirected graph.*

With the use of fast matrix multiplication, the algorithm's running time can be sped up. In particular, if $\omega = 2$ the running time becomes $\tilde{O}(n^{k/2+22/9(k+1)})$ for $k \geq 3$. With the current best bounds on ω and α, the algorithm runs in time

- $\tilde{O}(mn^{\omega/3})$ *for* $k = 2$,
- $\tilde{O}(n^{\frac{k}{2}+\frac{\beta_0}{k+1}+(\omega-2)})$ *for* $3 \leq k \leq 13$, *where* $\beta_0 := \frac{-8\omega^2+18\omega+18}{3\omega+3} \approx 1.5516$, *and*
- $\tilde{O}(n^{\frac{k}{2}+\frac{\beta_1}{k+1}+o(1)})$ *for* $k \geq 13$, *where* $\beta_1 := \frac{34\omega^2-24\omega-66}{3\omega+3} \approx 6.7533$.

We thus get that for every constant k, there is a $(2-\varepsilon, O(1))$-approximation algorithm running in almost $n^{k/2}$ time. Compare this with our conditional lower bound Theorem 3.13 that says that $n^{\lfloor k/2\rfloor - 1 - o(1)}$ time is needed for any $(9/5-\varepsilon, O(1))$-approximation.

As the approximation quality depends on k, we also present a version of our approximation scheme that trades-off running time for approximation quality.

THEOREM 1.5. *For any integer $1 \leq \ell \leq k$, there is a randomized combinatorial algorithm running in $\tilde{O}(mn + n^{1+k-\ell+\frac{\ell(\ell+1)}{2(k+1)}})$ time that computes a $\left(2-\frac{1}{2\ell}, 1-\frac{1}{2\ell}\right)$-approximation to the k-center of G w.h.p.*

This result appears in the text as Theorem 4.21. The running time of the algorithm can also be improved using fast matrix multiplication.

We note that our hardness result in Theorem 1.1 explains why k appears in the exponent of our running times: under SETH $n^{\Omega(k)}$ time is needed for any $(2-\varepsilon, O(1))$ approximation for constant $\varepsilon > 0$ independent of k.

Finally, we focus on the special case $k = 3$ for which we have stated two approximation algorithms so far. First, the $(3/2, 1/2)$-approximation algorithm of Theorem 1.2 runs in $O(n^{\omega+1/\omega})$ time. Second, our approximation scheme from Theorem 1.4 gives a $(9/5, 4/5)$-approximation with running time $O(n^{13/6})$ if $\omega = 2$. The latter is faster than the former (which would run in $O(n^{2.5})$ time if $\omega = 2$) but achieves a worse approximation ratio. As a minor final result, we show that one can obtain a mixed approximation with multiplicative factor between 3/2 and 9/5 (namely, 7/4) that runs faster than the $O(n^{\omega+1/\omega})$ time of the 3/2-approximation, and for a nontrivial set of graph sparsities m, runs faster than mn time even in weighted graphs. This result appears in the text as Theorem 4.26.

Related work. While we cannot hope to exhaustively list all the extensive work on k-center, here is some more. Approximation algorithms for the 1-center problem and the related diameter problem in graphs have been extensively studied [ACIM99, AGV23, CGR16, BRS$^+$21, AVW16, CLR$^+$14, RV13]. Dynamic $(2+\varepsilon)$-approximation algorithms for k-Center is a recent topic of interest [CFG$^+$24]. k-Center is studied in restricted classes of graphs and metrics both for static and for dynamic algorithms (e.g. [EKM14, DFHT03, Fel19b, FM20, GHL$^+$21, GG24] and many more). The asymmetric version of k-center (e.g. for directed graphs) is a harder problem, although 2-approximation algorithms are possible for structured metrics (see [BHW20] and the references therein).

2 Preliminaries

Let $G = (V, E)$ be a weighted or unweighted graph. Throughout this paper all graphs will be undirected. Denote by n the number of vertices in the graph $|V|$ and by m the number of edges $|E|$. The distance $d(u, v)$ between two vertices $u, v \in V$ is the length of the shortest path in G between u and v.

The eccentricity of a vertex v is defined as $\max_{u\in V} d(v, u)$. The vertex with smallest eccentricity is called the *center* of G and its eccentricity is the radius of G, $\min_{v\in V}\max_{u\in V} d(v, u)$. The k-center problem generalizes this definition to sets of k points. Define the k-radius of G, $R_k(G)$, to be

$$R_k(G) = \min_{\substack{C\subseteq V\\ |C|=k}} \max_{v\in V} d(v, C).$$

The k-center of G is defined as the set of points that achieve the k-radius,

$$\arg \min_{\substack{C\subseteq V\\ |C|=k}} \max_{v\in V} d(v, C).$$

When k is clear from context we refer to the k-radius as simply the radius.

Given a point $v \in V$ and $r > 0$ define the ball of radius r around v as $B(v, r) := \{u \in V : d(u, v) \leq r\}$. Similarly, for a set $S \subset V$, define the ball of radius r around the set S as $B(S, r) := \{u \in V : d(u, S) \leq r\}$. We often want to consider points that are far away from a point or set. For this we consider the complement of the ball around a point or set, $B(S, r)^c := \{u \in V : d(u, S) > r\}$. For ease of notation, we define $B(\emptyset, r) = \emptyset$ for any $r \in \mathbb{R}$.

The following values are defined in the arithmetic circuit model. The exponent ω is the smallest real number such that $n \times n$ matrices can be multiplied in $O(n^{\omega+\varepsilon})$ time for all $\varepsilon > 0$. The exponent $\omega(p, q, r)$ is the smallest real number such that one can multiply an $n^p \times n^q$ matrix by an $n^q \times n^r$ matrix in $O(n^{\omega(p,q,r)+\varepsilon})$ time for all $\varepsilon > 0$. It is known that $\omega(p,q,r)$ is invariant under any permutation of p,q,r. We also use the notation $\mathrm{MM}(a,b,c)$ to denote the best known running time to multiply an $a \times b$ matrix by a $b \times c$ matrix, in particular $\mathrm{MM}(n^p, n^q, n^r) = n^{\omega(p,q,r)}$. The value α is the largest value in $[0,1]$ such that $\omega(1,\alpha,1) = 2$.

3 Conditional Lower Bounds

In this section we will prove a simple lower bound for $(3/2-\varepsilon,\beta)$-approximating k-center (Theorem 3.6), and our main lower bound result for $(2-\varepsilon,\beta)$-approximation (Theorem 1.1), restated below with parameter k rescaled for convenience.

Theorem 3.1. (Equivalent form of Theorem 1.1) *There is a function $f\colon \mathbb{N}^+ \to \mathbb{N}^+$ such that the following holds. Let $k \geq 2$, $t, \ell \geq 1$ be constant integers. Assuming SETH, distinguishing between radius $\leq (2t+1)\ell$ and radius $\geq (4t+1)\ell$ for $(f(t)\cdot(k+1))$-center cannot be done in $O(m^{k-\delta})$ time, for any constant $\delta > 0$. Here, $f(t) = t$ for $t \in \{1,2,3,4\}$, and $f(t) \leq t^2$ for all $t \in \mathbb{N}^+$.*

Since our proof of the main result is quite involved, we will present the proof in an incremental fashion by first proving the simple lower bound (Theorem 3.6), and then two special cases (for $t = 1, 2$) of Theorem 3.1, before finally showing the full recursive construction for Theorem 3.1.

All our lower bound results hold for undirected unweighted graphs with n nodes and m edges (assuming $m \geq n-1$).

3.1 Known hardness results for Set Cover Our lower bounds rely on several hardness results for the Set Cover problem in the literature. A *Set Cover instance* is a bipartite graph $G = (A,B,E)$ on $n = |A| + |B|$ nodes, where we want to find the smallest subset $S \subseteq A$, such that every $b \in B$ is adjacent to some node in S. Without loss of generality, we assume each $b \in B$ is adjacent to at least one node A.

Our simple lower bound (proved in Section 3.2) is based on the following SETH-based hardness result for deciding whether a Set Cover instance has a size-k solution (or equivalently, solving the the k-Orthogonal-Vectors problem[3]).

Theorem 3.2. ([Wil05, PW10]) *Assuming SETH, for every integer $k \geq 2$ and $\delta > 0$, there is no $O(n^{k-\delta})$-time algorithm that can decide whether an n-node Set Cover instance (A,B,E) has a size-k solution, even when $|B| = O_\delta(k \log n)$.*

Starting from Section 3.3, our lower bounds will crucially rely on the inapproximability of Set Cover from the parameterized complexity literature [KLM19, Lin19].[4] These papers proved an inapproximability factor of $(\log n)^{\Omega(1)}$, while in our applications we only need (arbitrarily large) constant inapproximability factor C.

Lemma 3.3. (SETH-hardness of Gap Set Cover, implied by [KLM19, Theorem 1.5]) *Assuming SETH, for every integer $k \geq 2$ and for every $\delta > 0, C \geq 1$, no $O(n^{k-\delta})$-time algorithm can distinguish whether an n-node Set Cover instance $G = (A,B,E)$ has a solution of size k or has no solutions of size $\leq Ck$.*

For our purpose, we need the hardness to hold even for Set Cover instances (A,B,E) with small $|B|$. This can be obtained from Lemma 3.3 by a simple powering argument, as shown in the following corollary.

Corollary 3.4. (Small-B version of Lemma 3.3) *Assuming SETH, for every integer $k \geq 2$ and for every $\delta, \gamma > 0, C \geq 1$, no $O(n^{k-\delta})$-time algorithm can distinguish whether an n-node Set Cover instance $G' = (A', B, E')$ with $|B| \leq n^\gamma$ has a solution of size k or has no solutions of size $\leq Ck$.*

Proof. Let $g = \lceil 1/\gamma \rceil$. Given a Set Cover instance $G = (A,B,E)$, we can create a larger Set Cover instance $G' = (A^g, B, E')$, in which a g-tuple $(u_1,\dots,u_g) \in A^g$ is adjacent to $b \in B$ in E' iff there exists u_i such that

[3]A k-Orthogonal-Vectors (k-OV) instance is a set $A \subseteq \{0,1\}^d$ of n binary vectors of dimension d, and the k-OV problem asks if we can find k vectors $a_1,\dots,a_k \in A$ such that they are orthogonal, i.e., $(a_1)[i]\cdot(a_2)[i]\cdot\cdots\cdot(a_k)[i] = 0$ for all $i \in [d]$. Solving the k-OV instance $A \subseteq \{0,1\}^d$ is equivalent to deciding whether the Set Cover instance $G = (A,[d],E)$ defined by letting $(a,i) \in E$ iff $a[i] = 0$ has a size-k solution.

[4]These papers sometimes stated their results for the Dominating Set problem, which is essentially the same as Set Cover.

$(u_i, b) \in E$. Observe that for any nonnegative integer k', G has a size-$k'g$ solution if and only if G' has a size-k' solution.

By Lemma 3.3, it requires $(|A|+|B|)^{kg-o(1)}$ time to decide whether G has a size-kg solution or has no solutions of size $\leq Ckg$ under SETH. Hence, the same time is required for deciding whether G' has a size-k solution or has no solutions of size $\leq Ck$ under SETH. Note $G' = (A^g, B, E')$ has $|A|^g + |B|$ nodes, and we can pad dummy nodes into the A^g side so that G' now has $n = (|A|+|B|)^g$ nodes, and the number of nodes in B is only $|B| \leq n^{1/g} \leq n^{\gamma}$. The time lower bound then becomes $(|A|+|B|)^{kg-o(1)} = \Omega(n^{k-o(1)})$. □

We will also use the lower bound for Gap Set Cover under Exponential Time Hypothesis.

Lemma 3.5. (ETH-hardness of Gap Set Cover, implied by [KLM19, Theorem 1.4]) *Assuming ETH, for any constant $C \geq 1$, no $O(f(k)n^{o(k)})$-time algorithm can distinguish whether an n-node Set Cover instance $G = (A, B, E)$ has a solution of size k or has no solutions of size $\leq Ck$.*

3.2 A simple lower bound for $(\frac{3}{2} - \varepsilon, \beta)$-approximation

Our simple lower bound is stated as follows.

Theorem 3.6. *Let $k \geq 2$, $\ell \geq 1$ be constant integers. Assuming SETH, distinguishing between radius $\leq 2\ell$ and radius $\geq 3\ell$ for k-center cannot be done in $O(m^{k-\delta})$ time, for any constant $\delta > 0$.*

As a corollary, for any constants $\varepsilon \in (0,1), \beta \geq 0$, $(\frac{3}{2} - \varepsilon, \beta)$-approximating k-center radius requires $m^{k-o(1)}$ time under SETH.

To show the corollary, we set $\ell = \lfloor \frac{\beta}{2\varepsilon} \rfloor + 1$ which satisfies $(\frac{3}{2} - \varepsilon) \cdot 2\ell + \beta < 3\ell$, so a $(\frac{3}{2} - \varepsilon, \beta)$-approximation algorithm can distinguish between radius $\leq 2\ell$ and $\geq 3\ell$ and hence requires $m^{k-o(1)}$ time.

We prove Theorem 3.6 in the rest of this section.

Suppose we are given a Set Cover instance $G = (A, B, E)$ where $|A| = \Theta(n)$ and $|B| = n^{o(1)}$ and want to decide whether it has a size-k solution. By Theorem 3.2, this requires $n^{k-o(1)}$ time under SETH.

Based on the Set Cover instance, we define a base gadget graph (which will be repeatedly used in this section and later sections) as follows.

Definition 3.7. (Base gadget graph $\mathrm{Gad}(\hat{A}, \hat{B}, \hat{c}, L)$) *Let bipartite graph $G = (A, B, E)$ be the given Set Cover instance, and let $L \geq 1$ be an integer parameter. Then, the base gadget graph is defined by the following procedure:*

- *Create node sets $\hat{A}, \hat{B}$ which are copies of the node sets A, B. By convention, let $\hat{a} \in \hat{A}$ denote the copy of $a \in A$ (and similarly for $\hat{b} \in \hat{B}$ and $b \in B$).*
- *For every $(a, b) \in E$, connect $\hat{a} \in \hat{A}$ and $\hat{b} \in \hat{B}$ by an L-edge path $(\hat{a}, v_{\hat{a},\hat{b},1}, v_{\hat{a},\hat{b},2}, \ldots, v_{\hat{a},\hat{b},L-1}, \hat{b})$.*[5]
- *Add a new node $\hat{c}$.*
- *For every $\hat{a} \in \hat{A}$, connect $\hat{a}$ and $\hat{c}$ by an L-edge path $(\hat{a}, w_{\hat{a},1}, w_{\hat{a},2}, \ldots, w_{\hat{a},L-1}, \hat{c})$.*

We denote this base gadget graph by $\mathrm{Gad}(\hat{A}, \hat{B}, \hat{c}, L)$. (If we have a base gadget graph named $\mathrm{Gad}(A', B', c', L)$, then we will analogously use the convention that $a' \in A'$ denotes the copy of $a \in A$, and similarly for $b' \in B'$ and $b \in B$.)

Note that a Base gadget graph with parameter L has $L|E| + L|A| \leq O(L|A||B|) \leq Ln^{1+o(1)}$ edges.

Using the base gadget graph in Definition 3.7, we now create a k-center instance $G' = (V', E')$ as follows (see Fig. 1):

- Add a base gadget graph $\mathrm{Gad}(A', B', c', \ell)$.
- For every $b' \in B'$, attach an ℓ-edge path $(b', u_{b',1}, u_{b',2}, \ldots, u_{b',\ell})$ to b'.

Observe that the new graph G' has $O(\ell \cdot n^{1+o(1)})$ edges.

First we show a good k-center solution exists in the YES case.

[5]When we connect two nodes by a path, we mean we add new internal nodes and edges into the graph to form this path.

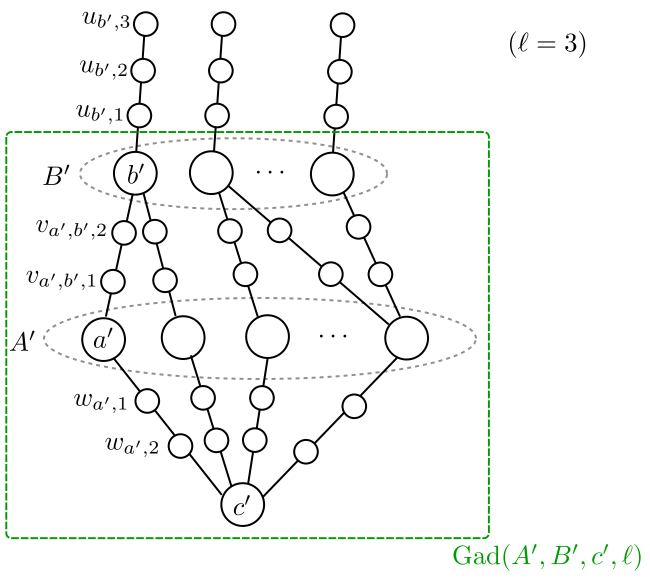

Figure 1: The k-center instance $G' = (V', E')$ constructed in the proof of Theorem 3.6 (for distinguishing radius $\leq 2\ell$ or $\geq 3\ell$). In this example, $\ell = 3$. The dashed green box contains the base gadget graph $\mathrm{Gad}(A', B', c', \ell)$. Given a Set Cover solution of size k, picking their copies in A' gives a k-center solution of radius 2ℓ.

Lemma 3.8. *If the original Set Cover instance $G = (A, B, E)$ has a solution $S \subseteq A$ of size $|S| = k$, then the k-center instance G' has a solution with radius 2ℓ.*

Proof. Given the Set Cover solution $S = \{s_1, \ldots, s_k\} \subset A$, in $G' = (V', E')$ we simply pick the copies of the same k nodes, $s'_1, \ldots, s'_k \in A' \subset V'$ as the centers, and we now verify that they cover every node in V' within distance 2ℓ:

- Every node $a' \in A' \subseteq V'$ can be reached from the center $s'_1 \in A'$ via a length-2ℓ path through node c', namely $(s'_1, w_{s'_1,1}, \cdots, w_{s'_1,\ell-1}, c', w_{a',\ell-1}, \cdots, w_{a',1}, a')$. This also means all intermediate nodes $w_{\cdot,\cdot}$ and the node c' can be reached from the center s'_1 within distance less than 2ℓ.
- For every $b \in B$, the Set Cover solution S guarantees that there exists an s_i such that $(s_i, b) \in E$. Then, by definition of the base gadget graph, $b' \in B' \subset V'$ can be reached from the center s'_i via a length-ℓ path $(s'_i, v_{s'_i,b',1}, \ldots, v_{s'_i,b',\ell-1}, b')$.

  Then, since all nodes $u_{b',\cdot}$ on the path attached to b', as well as all nodes $v_{\cdot,b',\cdot}$ on the paths between b' and A' in the base gadget graph, are reachable from b' within distance $\leq \ell$, they are hence reachable from s'_i within distance $\leq \ell + \ell = 2\ell$.

We have verified all nodes in V' are covered by some center s'_i within distance 2ℓ, finishing the proof. ◻

Next, we show no good k-center solution can exist in the NO case. To show this, we prove the contrapositive.

Lemma 3.9. *If the k-center instance G' has a solution with radius $< 3\ell$, then the original Set Cover instance $G = (A, B, E)$ has a solution of size k.*

Proof. Let $\{\tilde{s}_1, \ldots, \tilde{s}_k\} \subset V'$ denote the k-center solution of $G' = (V', E')$ of radius $< 3\ell$. For each $i \in [k]$, define s'_i to be the node in $A' \subset V'$ that is the closest to $\tilde{s}_i$ on graph G' (breaking ties arbitrarily). In other words,

- If $\tilde{s}_i \in \{u_{b',j}\}_{j\in[\ell]} \cup \{b'\}$ for some $b' \in B'$ (i.e., $\tilde{s}_i$ is on the ℓ-edge path attached to b'), then s'_i is the copy of some $s_i \in A$ such that $(s_i, b) \in E$ (which exists by our initial assumption that b has at least one neighbor in the Set Cover instance).

- If $\tilde{s}_i \in \{v_{a',b',j}\}_{b' \in B', j \in [\ell-1]} \cup \{a'\} \cup \{w_{a',j}\}_{j \in [\ell-1]}$ for some $a' \in A'$ (i.e., $\tilde{s}_i$ is in the base gadget graph excluding B' and $\{c'\}$), then $s'_i = a'$.

- If $\tilde{s}_i = c'$ then $s'_i \in A'$ is arbitrary (this case will not happen in our proof).

We now show that $\{s_1, \ldots, s_k\} \subset A$, namely the nodes corresponding to $s'_1, \ldots, s'_k \in A'$, is a Set Cover solution.

Fix any $b \in B$. Consider $u_{b',\ell}$, the last node on the ℓ-edge path attached to b', and suppose it is reachable from the center $\tilde{s}_i$ within distance $< 3\ell$. By inspecting the construction of G', observe that within $< 3\ell$ distance $u_{b',\ell}$ cannot reach node c' or any other $b'' \in B' \setminus \{b'\}$. Hence, such center $\tilde{s}_i$ within $< 3\ell$ distance from $u_{b',\ell}$ can only be one of the following two cases:

- Case 1: $\tilde{s}_i \in \{u_{b',j}\}_{j \in [\ell]} \cup \{b'\}$.

  By earlier discussion, we have $(s_i, b) \in E$ in this case.

- Case 2: $\tilde{s}_i \in \{v_{a',b'',j}\}_{b'' \in B', j \in [\ell-1]} \cup \{a'\} \cup \{w_{a',j}\}_{j \in [\ell-1]}$ for some $a' \in A'$.

  By earlier discussion, we have $s_i = a$ in this case.

  By inspecting the construction of G', we can observe that in this case we must have $(a, b) \in E$ (and hence $(s_i, b) \in E$). Intuitively speaking, the shortest path from $u_{b',\ell}$ to the center $\tilde{s}_i$ can exit b' only through some $v_{a',b',\ell-1}$ for which $(a, b) \in E$, and from there it cannot reach c' or any other $b'' \in B' \setminus \{b'\}$, so it always remains the closet to the same $a' \in A'$.

This finishes the proof that every $b \in B$ is covered by some s_i, so $\{s_1, \ldots, s_k\}$ is a Set Cover solution. $\square$

Proof of Theorem 3.6. By combining Lemma 3.8 and Lemma 3.9, we see that any algorithm on graphs of $m \le n^{1+o(1)}$ edges that distinguishes between k-center radius $\le 2\ell$ and $\ge 3\ell$ can be used to decide whether the original Set Cover instance has a size-k solution, which requires $n^{k-o(1)} \ge m^{k-o(1)}$ time under SETH. $\square$

3.3 Warm-up I: Lower bounds via Gap-Set-Cover

In this section, we prove the following special case of our full lower bound result Theorem 3.1, which achieves better inapproximability ratio $(5/3-\varepsilon)$ than the previous $(3/2-\varepsilon)$, but has a lower time lower bound $m^{k-1-o(1)}$. We prove this special case first in order to clearly illustrate how the hardness of Gap Set Cover is helpful, which is one of the main ideas behind our full result.

THEOREM 3.10. (SPECIAL CASE OF THEOREM 3.1) *Let $k \ge 2$, $\ell \ge 1$ be constant integers. Assuming SETH, distinguishing between radius $\le 3\ell$ and radius $\ge 5\ell$ for $(k+1)$-center cannot be done in $O(m^{k-\delta})$ time, for any constant $\delta > 0$.*

As a corollary, for any constants $\varepsilon \in (0,1), \beta \ge 0$, $(\frac{5}{3}-\varepsilon, \beta)$-approximating k-center radius requires $m^{k-1-o(1)}$ time under SETH.

We will use a similar construction as the simple lower bound (Theorem 3.6) from the previous section. In order to improve the ratio from $(3/2-\varepsilon)$ to $(5/3-\varepsilon)$, we would like to increase the parameter of the base gadget graph from ℓ to 2ℓ, while the paths attached to B' still have length ℓ. But naively doing this would make some parts of the graph no longer covered by the k chosen centers in A' within the desired radius 3ℓ. To solve this issue, we pick the node c' as an extra center to cover the remaining parts. In this way, a size-k Set Cover solution of implies a $(k+1)$-center solution of radius 3ℓ. And, in the converse direction, the same argument as before shows that a $(k+1)$-center solution of radius $< 5\ell$ would imply a $(k+1)$-size Set Cover solution. Therefore, we can use the hardness of Gap Set Cover (distinguishing between solution size $\le k$ or $> k+1$) to conclude the proof.

Proof of Theorem 3.10. Suppose we are given a Set Cover instance $G = (A, B, E)$ where $|A| = \Theta(n)$ and $|B| \le n^\gamma$ (where constant $\gamma > 0$ can be chosen arbitrarily small), and want to decide whether it has a size-k solution or has no solutions of size $\le k+1$. By Corollary 3.4, this requires $n^{k-o(1)}$ time under SETH.

We will create a $(k+1)$-center instance $G' = (V', E')$, in a similar way to the proof of Theorem 3.6:

- Add a base gadget graph $\mathrm{Gad}(A', B', c', 2\ell)$ (Definition 3.7).

- For every $b' \in B'$, attach an ℓ-edge path $(b', u_{b',1}, u_{b',2}, \ldots, u_{b',\ell})$ to b'.

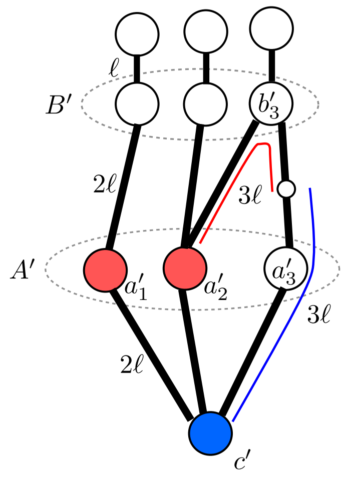

Figure 2: The $(k+1)$-center instance $G' = (V', E')$ constructed in the proof of Theorem 3.6 (for distinguishing radius $\leq 3\ell$ or $\geq 5\ell$). Here we use a thick segment to denote a path of prescribed number of edges. Given a Set Cover solution of size k (this example has $k = 2$ attained by $\{a_1, a_2\}$), picking their copies in A' together with c' gives a $(k+1)$-center solution of radius 3ℓ. In this example, on the 2ℓ-edge path from $a'_3 \in A'$ to $b'_3 \in B'$, the first half of nodes are covered by center c', and the second half are covered by center a'_2.

Observe that the new graph G' has $O(\ell|A||B|) \leq n^{1+\gamma}$ edges.

LEMMA 3.11. *If the original Set Cover instance $G = (A, B, E)$ has a solution $A' \subseteq A$ of size $|A'| = k$, then the $(k+1)$-center instance G' has a solution with radius 3ℓ.*

Proof. Given the Set Cover solution $S = \{s_1, \ldots, s_k\} \subset A$, in $G' = (V', E')$ we pick the copies of the same k nodes, $s'_1, \ldots, s'_k \in A' \subset V'$ together with $c' \in V'$ as the $(k+1)$ centers, and we now verify that they cover every node in V' within distance 3ℓ (see Fig. 2):

- As before, the Set Cover solution guarantees that every $b' \in B' \subset V'$ can be reached from some center s'_i within distance 2ℓ via the path $(s'_i, v_{s'_i,b',1}, \ldots, v_{s'_i,b',2\ell-1}, b')$.

  Then, the second half of the 2ℓ-edge paths between A' and b', namely the nodes $v_{\cdot,b',j}$ for $j \geq \ell$, are reachable from b' within distance $\leq \ell$, and hence reachable from that center within distance $\leq 2\ell + \ell = 3\ell$.

  Similarly, all nodes $u_{b',\cdot}$ on the ℓ-edge path attached to b' are also reachable from the center within distance 3ℓ.

- Observe that the center c' has distance $\leq 3\ell$ to the first half of the 2ℓ-edge paths between A', B', namely $v_{\cdot,\cdot,j}$ for $j \leq \ell$.

  Similarly one can check that c' covers all the remaining nodes in G' (including $c', w_{\cdot,\cdot}$, and nodes in A') within distance 3ℓ.

□

LEMMA 3.12. *If the $(k+1)$-center instance G' has a solution with radius $< 5\ell$, then the original Set Cover instance $G = (A, B, E)$ has a solution of size $(k+1)$.*

Proof Sketch. The proof of this lemma uses exactly the same argument as Lemma 3.9 from the previous section. The current lemma holds for radius $< 5\ell$ because the shortest distance from $u_{b',\ell}$ to node c' or any other $b'' \in B' \setminus \{b'\}$ is $\ell + 2\ell + 2\ell = 5\ell$. □

By combining the previous two lemmas, we see that any algorithm on graphs of $m \leq O(n^{1+\gamma})$ edges that distinguishes between $(k+1)$-center radius $\leq 3\ell$ and $\geq 5\ell$ can be used to decide whether the original Set Cover instance has a size-k solution or has no solutions of size $\leq k+1$, which requires $n^{k-o(1)} \geq m^{k/(1+\gamma)-o(1)}$ time. Since $\gamma > 0$ can be chosen arbitrarily small, we rule out $O(m^{k-\delta})$-time algorithms for all constant $\delta > 0$. □

3.4 Warm-up II: Recursively covering the paths In Section 3.3 we saw how using more centers can lead to higher inapproximability ratio. In this section we develop that idea further and prove the following result.

THEOREM 3.13. (SPECIAL CASE OF THEOREM 3.1) *Let $k \geq 2$, $\ell \geq 1$ be constant integers. Assuming SETH, distinguishing between radius $\leq 5\ell$ and radius $\geq 9\ell$ for $(2k+2)$-center cannot be done in $O(m^{k-\delta})$ time, for any constant $\delta > 0$.*

As a corollary, for any constants $\varepsilon \in (0,1), \beta \geq 0$, $(\frac{9}{5} - \varepsilon, \beta)$-approximating k-center radius requires $m^{\lfloor k/2 \rfloor - 1 - o(1)}$ time under SETH.

Compared to the previous construction in Section 3.3, here we will increase the inapproximability ratio by further increasing the parameter of the base gadget graph (namely the distance between A', B' and between A', c), which would again cause some of the A'-to-B' paths to be uncovered. This time we have to add more centers in order to cover everything: we will create another copy of the base gadget graph $\text{Gad}(\bar{A}, \bar{B}, \bar{c}, \bar{L})$ (for some smaller parameter $\bar{L}$), and additionally pick k centers from $\bar{A}$ (and $\bar{c}$) as well. We will connect edges appropriately so that the originally uncovered part on the A'-to-B' path between node $b' \in B'$ and any $a' \in A'$ will be covered by the new centers by going through $\bar{b} \in \bar{B}$.

We remark that our construction for proving Theorem 3.13 will be slightly redundant, and $(2k+2)$ in the theorem statement can in fact be improved to $(2k+1)$ (see Footnote 6). We present this slightly weaker version just to keep consistency with the full generalized construction to be described in the next section.

The rest of this section proves Theorem 3.13. Suppose we are given a Set Cover instance $G = (A, B, E)$ where $|A| = \Theta(n)$ and $|B| = n^{\gamma}$ (where constant $\gamma > 0$ can be chosen arbitrarily small), and want to decide whether it has a size-k solution or has no solutions of size $\leq 2k+2$. By Corollary 3.4, this requires $n^{k-o(1)}$ time under SETH.

We will create a $(2k+2)$-center instance $G' = (V', E')$ of $O(\ell|A||B|) \leq n^{1+\gamma}$ edges, as follows. See Fig. 3.

- Add two base gadget graphs $\text{Gad}(A', B', c', 4\ell)$ and $\text{Gad}(\bar{A}, \bar{B}, \bar{c}, 2\ell)$ (Definition 3.7).
- For every $b' \in B'$, attach an ℓ-edge path $(b', u_{b',1}, u_{b',2}, \dots, u_{b',\ell})$ to b'.
- Then, for every $(a, b) \in E$, let $v_{a',b',2\ell}$ denote the middle node on the 4ℓ-edge path between a' and b' in $\text{Gad}(A', B', c', 4\ell)$, and we add a 2ℓ-edge path that connects $v_{a',b',2\ell}$ and $\bar{b}$ (depicted as dashed blue paths in Fig. 3). Here we stress that both $\bar{b} \in \bar{B}$ and $b' \in B'$ are copies of the same $b \in B$.

LEMMA 3.14. *If the original Set Cover instance $G = (A, B, E)$ has a solution $S \subseteq A$ of size $|S| = k$, then the $(2k+2)$-center instance G' has a solution with radius 5ℓ.*

Proof. Given the Set Cover solution $S = \{s_1, \dots, s_k\} \subset A$, in $G' = (V', E')$ we pick both copies of the same k nodes, $s'_1, \dots, s'_k \in A', \bar{s_1}, \dots, \bar{s_k} \in \bar{A}$, and $c', \bar{c}$ as the $(2k+2)$ centers, and we now verify that they cover every node in V' within distance 5ℓ:

- Similarly to Lemma 3.11 from the previous section, here we see that within 5ℓ radius the centers $s'_1, \dots, s'_k \in A'$ together cover all the ℓ-edge paths attached to B', as well as the last $1/4$ fraction of every 4ℓ-edge path from A' to B'. Also, within 5ℓ radius, c' covers all the 4ℓ-edge paths between c' and A', as well as the first $1/4$ fraction of every 4ℓ-edge path from A' to B'.
- This is the key part in our construction: For every 4ℓ-edge path between $a' \in A$ and $b' \in B'$, it remains to verify that its middle $1/2$ fraction (namely the nodes $v_{a',b',j}$ where $\ell < j < 3\ell$) are also covered within 5ℓ radius, for which it is sufficient to show that the middle node $v_{a',b',2\ell}$ can reach some center within 4ℓ distance. To show this, suppose b is covered by $s_i \in S$ in the Set Cover solution, and note that the center $\bar{s_i} \in \bar{A}$ can reach the middle node via a 4ℓ-edge path $\bar{s_i} \overset{2\ell}{\rightsquigarrow} \bar{b} \overset{2\ell}{\rightsquigarrow} v_{a',b',2\ell}$. (This also shows that the intermediate nodes on the path $\bar{b} \overset{2\ell}{\rightsquigarrow} v_{a',b',2\ell}$ are covered.)
- Then, observe that the remaining nodes in the graph, namely the nodes on the 2ℓ-edge paths between $\bar{A}, \bar{B}$ and between $\bar{c}, \bar{A}$, are covered by the center $\bar{c}$ within $4\ell < 5\ell$ distance.[6]

[6]Alternatively, one can show that they are covered by the centers $\bar{s}_1, \dots, \bar{s}_k$ within 4ℓ distance. Hence it is actually not necessary to include $\bar{c}$ as a center.

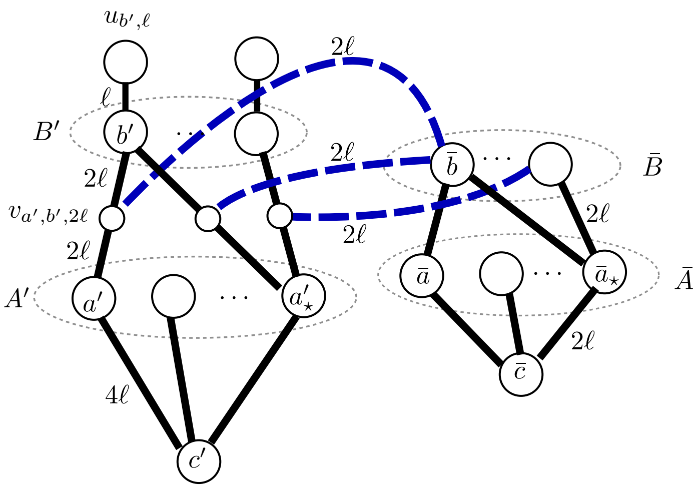

Figure 3: The $(2k + 2)$-center instance $G' = (V', E')$ constructed in the proof of Theorem 3.13 (distinguishing between radius $\leq 5\ell$ or $\geq 9\ell$). For example, the 4ℓ-edge path between a' and b' can be covered within radius 5ℓ by centers $a'_\star, \bar{a}_\star, c'$.

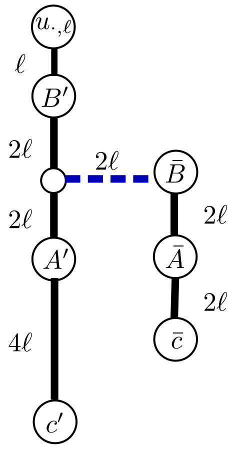

Figure 4: The "skeleton graph" of the graph $G' = (V', E')$ in Fig. 3. Here, each super-node represents a subset of nodes in G', and the lengths denote shortest distances between subsets of nodes.

□

Now we proceed to the NO case. For the sake of analysis, we introduce a few terminologies. We *associate* each node in the $(2k+2)$-center instance G' to at most one node in A, and to at most one node in B in the most natural way. Also, some of the nodes in G' are called *type-A* (or *type-B*, *type-C*). Their precise definitions are given as follows (we state the definitions with a little bit of generality so that it can be reused in the next section where G' may contain even more copies of the base gadget graph):

Definition 3.15. *In each copy of the base gadget graph* $\mathrm{Gad}(\hat{A}, \hat{B}, \hat{c}, L)$ *(Definition 3.7):*

- *Each $\hat{a} \in \hat{A}$ is associated to $a \in A$ (recall that $\hat{a}$ is a copy of a), and we say $\hat{a}$ is a type-A node.*
- *Similarly, each $\hat{b} \in \hat{B}$ is associated to $b \in B$, and we say $\hat{b}$ is a type-B node.*
- *$\hat{c}$ is not associated to any node. We say $\hat{c}$ is a type-C node.*
- *For each L-edge path between $\hat{a} \in \hat{A}$ and $\hat{b} \in \hat{B}$, all internal nodes on this path (namely, $v_{\hat{a},\hat{b},j}, j \in [L-1]$) are associated to both $a \in A$ and $b \in B$.*
- *For each L-edge path between $\hat{a} \in \hat{A}$ and $\hat{c}$, all internal nodes on this path (namely, $w_{\hat{a},j}, j \in [L-1]$) are associated to $a \in A$.*

Then, in the main construction of $G' = (V', E')$:

- *For each ℓ-edge path $(b', u_{b',1}, u_{b',2}, \ldots, u_{b',\ell})$ attached to $b' \in B'$, all nodes on this path are associated to $b \in B$.*
- *Whenever we add a path from some $v_{\hat{a},\hat{b},j}$ inside a base gadget graph $\mathrm{Gad}(\hat{A}, \hat{B}, \hat{c}, L)$ to another node outside this base gadget graph, we associate all internal nodes on this added path to both $a \in A$ and $b \in B$. (In the example in Fig. 3, these paths are depicted as dashed blue paths.)*

Now we observe a few simple but useful properties of the constructed instance $G' = (V', E')$ and the way we associated nodes of V' to nodes of A, B:

Observation 3.16. *If a node in G' is associated to both $a \in A$ and to $b \in B$, then $(a, b) \in E$ in the original Set Cover instance.*

Observation 3.17. *For every edge (x, y) in G':*

1. *("Remember a"): If x is associated to some $a_x \in A$ and y is associated to some $a_y \in A$, then $a_x = a_y$.*
2. *("Forget a"): If x is associated to some $a_x \in A$, and y is not associated to any $a \in A$, then y is a type-B or type-C node.*
3. *("Remember b"): If x is associated to some $b_x \in B$ and y is associated to some $b_y \in B$, then $b_x = b_y$.*
4. *("Forget b but still remember a"): If x is associated to some $b_x \in B$, and y is not associated to any $b \in B$, then y is a type-A node, and y is associated to an $a_y \in A$ such that $(a_y, b_x) \in E$ in the Set Cover instance.*

Both observations can be directly verified by carefully examining Definition 3.7, our definition of $G' = (V', E')$ (see Fig. 3), and Definition 3.15.

Now we can prove the lemma for the NO case:

Lemma 3.18. *If the $(2k + 2)$-center instance G' has a solution with radius $< 9\ell$, then the original Set Cover instance $G = (A, B, E)$ has a solution of size $2k + 2$.*

Proof. Let $\{\tilde{s}_1, \dots, \tilde{s}_{2k+2}\} \subset V'$ denote the $(2k + 2)$-center solution of $G' = (V', E')$ of radius $< 9\ell$. We define $\{s_1, \dots, s_{2k+2}\} \subset A$ as follows.

Definition 3.19. *Given a center $\tilde{s}_i \in V'$, we define $s_i \in A$ as follows:*

- *If $\tilde{s}_i$ is associated to some $a \in A$ (see Definition 3.15), then let $s_i = a$.*
- *If $\tilde{s}_i$ is associated to $b \in B$ but not to any node in A, then take an arbitrary $a \in A$ such that $(a, b) \in E$ (which exists by our initial assumption of the Set Cover instance) and let $s_i = a$.*
- *If $\tilde{s}_i$ is not associated to any node in A or B, then just let s_i be an arbitrary node in A.*

We will show that $\{s_1, \dots, s_{2k+2}\} \subset A$ is a Set Cover solution. Fix any $b \in B$, and consider $u_{b',\ell}$ (the last node on the ℓ-edge path attached to $b' \in B'$). There is a center $\tilde{s}_i$ at distance $< 9\ell$ from $u_{b',\ell}$. It remains to prove that $(s_i, b) \in E$ in the Set Cover instance.

Let P denote the shortest path from $u_{b',\ell}$ to the center $\tilde{s}_i$ of length $|P| < 9\ell$. We make the following definition.

Definition 3.20. *We say a path $P = (p_0, p_1, \dots, p_{|P|})$ is* bad*, if there exists $0 \le x < y \le |P|$ such that p_x is a type-A node, and p_y is a type-B or type-C node.*

Then we will prove the following two lemmas, which together with $|P| < 9\ell$ immediately imply $(s_i, b) \in E$ as desired, finishing the proof of Lemma 3.18.

Lemma 3.21. *In G', any bad path P starting from $u_{b',\ell}$ must have length $\ge 9\ell$.*

Lemma 3.22. *If there is a path P from $u_{b',\ell}$ to $\tilde{s}_i$ that is not bad, then s_i (as defined in Definition 3.19) satisfies $(s_i, b) \in E$ in the Set Cover instance.*

Proof of Lemma 3.21. In our construction of G', type-A nodes are $A' \cup \bar{A}$, type-B nodes are $B' \cup \bar{B}$, and type-C nodes are $\{c', \bar{c}\}$. In a bad path $P = (p_0, \dots, p_{|P|})$ where $p_0 = u_{b',\ell}$, suppose p_x is a type-A node and p_y is a type-B or type-C node ($0 \le x < y \le |P|$). We now use a case distinction. (The distance lower bounds we are using here can be seen more transparently from the "skeleton graph" of our construction depicted in Fig. 4.)

- Case $p_x \in A'$: Observe $x \ge d_{G'}(u_{b',\ell}, A') = 5\ell$, and $y - x \ge d_{G'}(A', \{c', \bar{c}\} \cup B' \cup \bar{B}) = 4\ell$.
- Case $p_x \in \bar{A}$: Observe $x \ge d_{G'}(u_{b',\ell}, \bar{A}) = 7\ell$, and $y - x \ge d_{G'}(\bar{A}, \{c', \bar{c}\} \cup B' \cup \bar{B}) = 2\ell$.

Algorithm 1 Recursive construction for proving Theorem 3.1

```
 1: Global Input: A Set Cover instance $G = (A, B, E)$ and constant integers $t, \ell \geq 1$.
 2:
 3: procedure Recurse($p$)
 4: Input: positive integer $1 \leq p \leq 2t$
 5: Output: a tuple $(\bar{G}, \bar{A}, \bar{B})$, where $\bar{G} = (\bar{V}, \bar{E})$ is a graph, and $\bar{A}, \bar{B} \subseteq \bar{V}$ are copies of $A, B$
 6:     $\bar{G} \leftarrow$ a fresh copy of the base gadget graph $\mathrm{Gad}(\bar{A}, \bar{B}, \bar{c}, (2t + 1 - p)\ell)$
 7:     for $q \in \{1, 2, \ldots, \lfloor \frac{2t-p}{2p} \rfloor\}$ do            ▷ Non-empty iff $p \leq 2t/3$
 8:         $(G'', A'', B'') \leftarrow$ Recurse($(2q + 1) \cdot p$)                  ▷ $(2q + 1)p \leq 2t$ holds
 9:         Add $G''$ into $\bar{G}$
10:         for $(a, b) \in E$ do
11:             Add a $2qp\ell$-edge path in $\bar{G}$ between $v_{\bar{a},\bar{b},(2t+1-p-2qp)\ell} \in \mathrm{Gad}(\bar{A}, \bar{B}, \bar{c}, (2t + 1 - p)\ell)$ and $b'' \in B''$   ▷
    By convention, $\bar{b} \in \bar{B}, b'' \in B''$ are copies of $b \in B$
12:         end for
13:     end for
14:     return $(\bar{G}, \bar{A}, \bar{B})$
15: end procedure
16:
17: procedure Main
18:     $(G', A', B') \leftarrow$ Recurse(1)
19:     For every $b' \in B'$, attach an $\ell$-edge path $(b', u_{b',1}, u_{b',2}, \ldots, u_{b',\ell})$ to $b'$.
20:     return $G'$
21: end procedure
```

In both cases we have $|P| \geq y = x + (y - x) \geq 9\ell$. □

Proof of Lemma 3.22. Let $P = (p_0, p_1, \ldots, p_{|P|})$. Pick the smallest $x \in [0, |P|]$ such that p_x is not associated to any node in B (if none exists, let $x = |P| + 1$). Since $p_0 = u_{b',\ell}$ is associated to b, by repeatedly applying Item 3 of Observation 3.17 we know p_{x-1} is also associated to b. If $x = |P| + 1$, then this means $p_{|P|} = \tilde{s}_i$ is associated to b, and then from Definition 3.19 and Observation 3.16 we have $(s_i, b) \in E$ as claimed. So we assume $x \leq |P|$ from now on.

Since p_{x-1} is associated to $b \in B$ but p_x is not associated to any node in B, by Item 4 of Observation 3.17 we know p_x is a type-A node and is associated to some $a \in A$ such that $(a, b) \in E$. Pick the smallest $y \in [x, |P|]$ such that p_y is not associated to any node in A (if none exists, let $y = |P| + 1$). Then, by repeatedly applying Item 1 of Observation 3.17 we know p_{y-1} is also associated to the same a. If $y = |P| + 1$, then this means $p_{|P|} = \tilde{s}_i$ is associated to the same $a \in A$ satisfying $(a, b) \in E$, and we have $s_i = a$ by Definition 3.19, and hence $(s_i, b) \in E$ as claimed. So we assume $y \leq |P|$ from now on.

Since p_{y-1} is associated to $a \in A$ but p_y is not associated to any node in A, by Item 2 of Observation 3.17 we know p_y is a type-B or type-C node. Hence, we have found $0 \leq x < y \leq |P|$ witnessing that P is a bad path, contradicting the assumption that P is not bad. □

□

Proof of Theorem 3.13. By combining Lemma 3.14 and Lemma 3.18, we see that any $(2k + 2)$-center algorithm on graphs of $m \leq O(n^{1+\gamma})$ edges that distinguishes between radius $\leq 5\ell$ and $\geq 9\ell$ can be used to decide whether the original Set Cover instance has a size-k solution or has no solutions of size $\leq 2k + 2$, which requires $n^{k-o(1)} \geq m^{k/(1+\gamma)-o(1)}$ time. Since $\gamma > 0$ can be chosen arbitrarily small, we rule out $O(m^{k-\delta})$-time algorithms for all constant $\delta > 0$. □

3.5 The full recursive construction for $(2 - \varepsilon, \beta)$-inapproximability

In this section we prove our full hardness result Theorem 3.1. The proof generalizes the construction from the previous section (Theorem 3.13) by adding even more copies of the base gadget graph, and hence achieves better inapproximability ratio. More specifically:

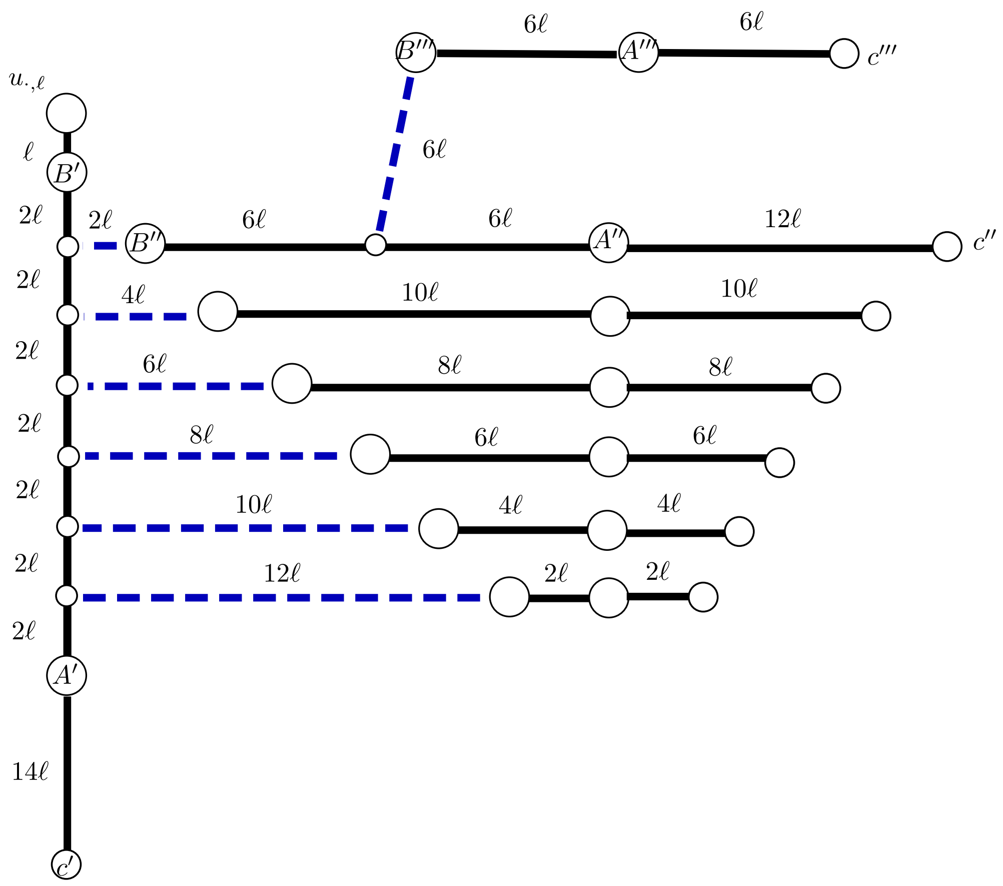

Figure 5: The "skeleton graph" of graph G' constructed by Algorithm 1 for $t = 7$ (for distinguishing between radius $\le 15\ell$ or $\ge 29\ell$ for $f(7)\cdot(k+1)$-center). To avoid clutter, we do not draw the graph G' itself; one can refer to Fig. 3 and Fig. 4 to understand the relation between G' and its skeleton graph by analogy.

- In the proof of Theorem 3.13 we only had one junction point on each 4ℓ-edge path between A' and B'. Now we will put more junction points (in order to cover a bigger fraction of the path).

- The proof of Theorem 3.13 only had one level of recursion, but in general we may continue the recursion by, for example, adding junction points on the paths between $\bar{A}$ and $\bar{B}$ and connect them to another copy of the base gadget graph.

Throughout, let $t, \ell \ge 1$ be fixed integers, and suppose we are given a Set Cover instance $G = (A, B, E)$ where $|A| = \Theta(n)$ and $|B| \le n^\gamma$ (where constant $\gamma > 0$ can be chosen arbitrarily small). We describe the general construction of graph $G' = (V', E')$ as a recursive procedure in Algorithm 1. The recursion is parameterized by a positive integer variable p. We remark that the constructions from Section 3.3 and Section 3.4 can be obtained from running Algorithm 1 with $t = 1$ and $t = 2$, respectively. For reference, in Fig. 5 we include the "skeleton graph" of the graph G' returned by Algorithm 1 for $t = 7$.

Let $h(p)$ denote the total number of base gadget graphs contained in the graph returned by RECURSE(p). The final graph G' returned by Algorithm 1 contains $h(1)$ copies of base gadget graphs ($h(1)$ is a function of t), and in total $O(h(1)\cdot|A||B|t\ell) \le O(n^{1+\gamma})$ many edges and nodes. We will analyze the dependence of $h(1)$ on t in the end of this section.

As in the previous section, we use the same rule as Definition 3.15 to associate each node in the constructed instance G' to at most one node in A and at most one node in B, and use the same definition of type-A (type-B, type-C) nodes as Definition 3.15. We use the same definition of bad paths as Definition 3.20.

By inspecting our recursive construction, we can see that both Observation 3.16 and Observation 3.17 still hold for the constructed G'.

We inductively prove the following properties of the recursive construction.

Lemma 3.23. *Suppose* Recurse(p) *returns* $(\bar{G}, \bar{A}, \bar{B})$. *Then both of the following hold.*

1. *Any bad path in* $\bar{G}$ *starting from any* $\bar{b} \in \bar{B}$ *must have length at least* $\geq (4t+2-2p)\ell$.

2. *Suppose the original Set Cover instance has a size-*k *solution. If we pick the copies of these* k *nodes in* $\hat{A}$ *and node* $\hat{c}$ *in all base gadget graphs* $\mathrm{Gad}(\hat{A}, \hat{B}, \hat{c}, L)$ *in* $\bar{G}$ *as centers (picking* $h(p) \cdot (k+1)$ *centers in total), then all nodes in* $\bar{G}$ *are covered within radius* $(2t+1)\ell$. *Moreover, all nodes in* $\bar{B}$ *are covered within radius* $(2t+1-p)\ell$.

Proof of Lemma 3.23, Item 1. Take the shortest bad path $P = (p_0, \ldots, p_{|P|})$ in $\bar{G}$ starting from any $p_0 = \bar{b} \in \bar{B}$, and let p_x be the first type-A node on P. It ends at some type-B or type-C node $p_{|P|}$ (since otherwise P has a proper prefix that is also bad). There are two cases:

- Case 1: $p_x \in \bar{A}$.

  We have $x \geq d_{\bar{G}}(\bar{B}, \bar{A}) = (2t+1-p)\ell$ by Line 6. The type-B or type-C node $p_{|P|}$ is either from $\bar{B} \cup \{\bar{c}\}$, or from some subgraph G'' constructed in a recursive call to Recurse$((2q+1) \cdot p)$ at Line 8. In the former case, observe that $|P| - x \geq d_{\bar{G}}(\bar{A}, \bar{B} \cup \{\bar{c}\}) = (2t+1-p)\ell$ (by Lines 6). In the latter case, from Line 11 we observe that $|P| - x \geq d_{\bar{G}}(\bar{A}, B'') = (2t+1-p-2qp)\ell + 2pq\ell = (2t+1-p)\ell$. In either case we get $|P| = x + (|P| - x) \geq (4t+2-2p)\ell$.

- Case 2: $p_x \notin \bar{A}$.

  Then, p_x can only be some type-A node from some subgraph G'' constructed in a recursive call to Recurse$((2q+1) \cdot p)$ at Line 8. Note that the path from p_0 to p_x has to visit B'' in order to enter G''. Let p_j be the maximum $j < x$ such that $p_j \in B''$. Then the path $(p_j, \ldots, p_x)$ must be entirely in G''. Note that the path $(p_x, \ldots, p_{|P|})$ is also entirely in G'', since otherwise it has to exit G'' through some $p_y \in B''$, but in that case the path $(p_0, \ldots, p_x, \ldots, p_y)$ is a shorter bad path. Hence, we have argued that the bad path $(p_j, \ldots, p_x, \ldots, p_{|P|})$ starting from $p_j \in B''$ is entirely in G'', and hence by the inductive hypothesis it must have at length $|P| - j \geq (4t+2-2(2q+1) \cdot p)\ell$. On the other hand, from Line 11 observe that $j = d_{\bar{G}}(\bar{b}, p_j) \geq d_{\bar{G}}(\bar{B}, B'') = \big((2t+1-p)\ell - (2t+1-p-2qp)\ell\big) + 2qp\ell = 4qp\ell$. So $|P| = j + (|P| - j) \geq (4t+2-2p)\ell$.

∎

Proof of Lemma 3.23, Item 2. Inside $\mathrm{Gad}(\bar{A}, \bar{B}, \bar{c}, (2t+1-p)\ell)$ added at Line 6, within radius $(2t+1)\ell$, center $\bar{c}$ covers all the $(2t+1-p)\ell$-edge paths adjacent to $\bar{c}$, and also covers all $v_{\bar{a},\bar{b},j}$ where $1 \leq j \leq (2t+1)\ell - (2t+1-p)\ell = p\ell$. Also, the Set Cover solution guarantees that picking the k centers in $\bar{A}$ can cover all nodes in $\bar{B}$ within radius $(2t+1-p)\ell$ (this proves the "Moreover" part), and hence cover all $v_{\bar{a},\bar{b},j}$ where $j \geq (2t+1-2p)\ell$ within radius $(2t+1)\ell$.

The remaining nodes on the $\bar{A}$-to-$\bar{B}$ paths are those $v_{\bar{a},\bar{b},j}$ with $p\ell < j < (2t+1-2p)\ell$, and we can check that each of them is at distance $\leq p\ell$ to a junction node $v_{\bar{a},\bar{b},(2t+1-p-2qp)\ell}$ for some $q \in \{1, 2, \ldots, \lfloor \frac{2t-p}{2p} \rfloor\}$, which is connected via a $2qp\ell$-edge path to $b'' \in B''$ (by Line 11). Since (G'', A'', B'') is returned from Recurse$((2q+1) \cdot p)$, by the inductive hypothesis, b'' is covered by some center within radius $(2t+1-(2q+1) \cdot p)\ell$, so this center covers $v_{\bar{a},\bar{b},(2t+1-p-2qp)\ell}$ (and all nodes on that $2qp\ell$-edge path) within radius $(2t+1-(2q+1) \cdot p)\ell + 2qp\ell = (2t+1-p)\ell$. Hence, all remaining nodes $v_{\bar{a},\bar{b},j}$ (where $p\ell < j < (2t+1-2p)\ell$) are covered within radius $(2t+1-p)\ell + p\ell = (2t+1)\ell$.

By the inductive hypothesis, all nodes in the recursively created subgraphs G'' are also covered within radius $(2t+1)\ell$. Hence, we have verified that all nodes in $\bar{G}$ are covered within radius $(2t+1)\ell$. ∎

Now we get the corollary for the top level of the recursion.

Corollary 3.24. *Suppose Algorithm 1 returns* (G', A', B') *(where* G' *contains* $h(1)$ *copies of the basic gadget graph). Then both of the following hold.*

1. *Any bad path in G' starting from any $u_{b',\ell}$ (defined in Line 19) must have length at least $\geq (4t+1)\ell$.*

2. *Suppose the original Set Cover instance has a size-k solution. We can cover all nodes in G' within radius $(2t+1)\ell$ by picking $h(1)\cdot(k+1)$ centers.*

Proof. For Item 1, note that in order for the bad path P from $u_{b',\ell}$ to reach any type-A node, it must first traverse the ℓ-edge path $(u_{b',\ell},\dots,u_{b',1},b')$. Since none of the traversed nodes so far are type-A nodes, by the definition of bad paths (Definition 3.20) we know the suffix of P starting from b' is still a bad path. By Lemma 3.23 Item 1 (for $p=1$), this suffix path must have length at least $(4t+2-2)\ell = 4t\ell$. Hence $|P| \geq \ell + 4t\ell = (4t+1)\ell$.

For Item 2, note that every $b' \in B'$ is covered within radius $(2t+1-1)\ell = 2t\ell$ by Lemma 3.23 Item 2, and hence all nodes on the ℓ-edge path attached to b (at Line 19) are covered within $(2t+1)\ell$ radius. All the remaining nodes in G' are also covered within $(2t+1)\ell$ radius by Lemma 3.23 Item 2. ∎

Corollary 3.24 Item 2 already deals with the YES case. Now we can use Corollary 3.24 Item 1 with the same arguments as before to analyze the NO case.

Corollary 3.25. *If the graph G' constructed by Algorithm 1 has a $h(1)\cdot(k+1)$-center solution with radius $< (4t+1)\ell$, then the original Set Cover instance $G=(A,B,E)$ has a solution of size $h(1)\cdot(k+1)$.*

Proof. The proof uses the same argument as Lemma 3.18 from previous section. Let $\{\tilde{s}_1,\dots,\tilde{s}_{h(1)\cdot(k+1)}\} \subset V'$ denote the $(h(1)\cdot(k+1))$-center solution of $G'=(V',E')$ of radius $< (4t+1)\ell$. We define $\{s_1,\dots,s_{h(1)\cdot(k+1)}\} \subset A$ using Definition 3.19. For any $b' \in B'$, consider $u_{b',\ell}$ (the last node on the ℓ-edge path attached to $b' \in B'$), and consider its shortest path P to a center $\tilde{s}_i$ at distance $< (4t+1)\ell$. By Corollary 3.24 Item 1, P cannot be a bad path. Then by Lemma 3.22, we have $(s_i, b) \in E$ in the Set Cover instance, finishing the proof that $\{s_1,\dots,s_{h(1)\cdot(k+1)}\} \subset A$ is a Set Cover solution. ∎

Proof of Theorem 3.1. By combining Corollary 3.24 Item 2 and Corollary 3.25, we see that any $(h(1)(k+1))$-center algorithm on graphs of $m \leq O(n^{1+\gamma})$ edges that distinguishes between radius $\leq (2t+1)\ell$ and $\geq (4t+1)\ell$ can be used to decide whether the original Set Cover instance has a size-k solution or has no solutions of size $\leq (h(1)(k+1))$, which requires $n^{k-o(1)} \geq m^{k/(1+\gamma)-o(1)}$ time by Corollary 3.4. Since $\gamma > 0$ can be chosen arbitrarily small, we rule out $O(m^{k-\delta})$-time algorithms for all constant $\delta > 0$. Here, $h(1) = f(t)$ is a function of t. The last lemma gives a (loose) upper bound on $f(t)$.

Lemma 3.26. $f(t) = h(1) \leq \begin{cases} t & 1 \leq t \leq 4, \\ 6 & t = 5, \\ t^2 & \textit{otherwise.} \end{cases}$

Proof. By Algorithm 1 we have the following recurrence:

$$h(p) = \begin{cases} 1 & 2t/3 < p \leq 2t, \\ 1 + \sum_{q=1}^{\lfloor (2t-p)/(2p) \rfloor} h((2q+1)\cdot p) & 1 \leq p \leq 2t/3. \end{cases} \tag{3.1}$$

Equivalently, $h(1)$ counts the number of integer sequences $(p_1,\dots,p_{s-1},p_s)$ (with flexible length $s \geq 1$) where $p_1 = 1$, $p_s \leq 2t$ and $p_i/p_{i-1} \in \{3,5,7,9,\dots\}$. As an alternative way to count these sequences, let $g(T)$ denote the number of sequences $(p_1,\dots,p_{s-1},p_s)$ (with flexible length $s \geq 1$) where $p_1 = 1$, $p_i/p_{i-1} \in \{3,5,7,9,\dots\}$, and $p_s \leq T$. Then $h(1) = g(2t)$, and g has the recurrence

$$g(T) = \begin{cases} 1 + g(\lfloor T/3 \rfloor) + g(\lfloor T/5 \rfloor) + g(\lfloor T/7 \rfloor) + g(\lfloor T/9 \rfloor) + \dots & T \geq 1 \\ 0 & T = 0. \end{cases} \tag{3.2}$$

Here are values of $g(T)$ for some small T: $g(0) = 0$, $g(1) = g(2) = 1, g(3) = g(4) = 2, g(5) = g(6) = 3, g(7) = g(8) = 4, g(9) = g(10) = 6$.

We will not attempt to study the asymptotics of $g(T)$. Instead, we prove a very loose upper bound $g(T) \leq \max\{1, 0.25T^2\}$ by induction: For $T \leq 10$, this inequality holds. For $T \geq 11$, we have $g(\lfloor T/(2k+1) \rfloor) = 1$

if $T/(2k+1) \in [1,3)$, and otherwise $g(\lfloor T/(2k+1) \rfloor) \le 0.25T^2/(2k+1)^2$ by the inductive hypothesis. Hence,

$$\begin{aligned} g(T) &\le 1 + |\{k \in \mathbb{N}^+ : T/(2k+1) \in [1,3)\}| + \sum_{k\in\mathbb{N}^+} 0.25T^2/(2k+1)^2 \\ &\le T/2 + 0.25T^2 \cdot 0.234 \\ &< 0.25T^2. \end{aligned}$$

Hence, $f(t) = g(2t) \le t^2$. □

□

Finally, we remark that if we assume ETH instead of SETH, we can use the same proof as above (but using the ETH-hardness of Gap Set Cover Lemma 3.5 instead) and show that there can be no $f(k)n^{o(k)}$ time $(2-\varepsilon,\beta)$-approximation algorithm for k-Center for $\varepsilon > 0, \beta = O(1)$.

4 Improved Approximation Algorithms for k-Center

We now turn to our upper bounds and show that if we allow a small additive error, we are able to break the 2-approximation barrier for k-center. In fact, we achieve a < 2 multiplicative approximation for any unweighted graph with k-radius > 1. We begin by showing an approximation algorithm for 2-center that obtains a multiplicative error of 5/3 and additive error of $\le 2/3$. We then construct a general algorithm for any k that achieves a $(3/2, 1/2)$-approximation. Next we construct a faster algorithm that achieves an approximation of $\left(2 - \frac{1}{2k-1}, 1 - \frac{1}{2k-1}\right)$. We then show how to interpolate between these two algorithms and obtain a $\left(2 - \frac{1}{2\ell}, 1 - \frac{1}{2\ell}\right)$ approximation for any $\ell < k$. We conclude with an improved algorithm for 3-center, which allows for bounded integer edge weights.

4.1 Warm Up - 5/3 Approximation for 2-Center We begin by presenting the first < 2 approximation algorithm for k-center running in $o(n^k)$ time (in sparse graphs), a 5/3 approximation algorithm for 2-center running in $O(mn^{\omega/3})$ time, which incurs an additive error when the radius is not divisible by 3. Our algorithm runs in two steps, both of which we generalize later to use in k-center approximation for any k. We prove the following statement.

Theorem 4.1. *There exists an algorithm running in $\tilde{O}(mn^{\omega/3})$ that computes a $(5/3, 2/3)$-approximation to the 2-center of any undirected, unweighted graph, w.h.p. If the 2-center radius is divisible by 3, the algorithm gives a true multiplicative 5/3-approximation.*

Proof. We construct an algorithm that is given an integer R that is divisible by 3 and determines whether $R_2(G) > R$ or $R_2(G) \le \frac{5R}{3}$. By performing a binary search for the correct value of R, we obtain a multiplicative 5/3 approximation if the 2-radius of G is divisible by 3. If R is not divisible by 3, the following algorithm can instead determine if $R_2(G) > R$ or $R_2(G) \le 2R - \left\lfloor \frac{R}{3} \right\rfloor$. This gives an additional additive error of at most $\frac{2}{3}$. For the sake of readability, we will only prove the case when R is divisible by 3, and note that by replacing $\frac{2R}{3}$ with either $R - \left\lfloor \frac{R}{3} \right\rfloor$ or $2\left\lfloor \frac{R}{3} \right\rfloor$ we obtain the general case.

Assume $R_2(G) \le R$, in this case we show that we are able to find a pair of points that cover all vertices with radius $\frac{5R}{3}$. If we are unable to find such a pair, we output $R_2(G) > R$.

Let c_1, c_2 be the optimal 2-centers, so that for every $v \in V$, either $d(c_1, v) \le R$ or $d(c_2, v) \le R$. Set $\delta = (2\omega - 3)/(3\omega - 3)$ and sample a random subset S of V of size $O(n^{1-\delta}\log n)$. In $\tilde{O}(mn^{1-\delta})$ time compute all distances between every $s \in S$ and $v \in V$.

Let w be the furthest node from the set S. Run BFS from w and let W be the closest n^δ nodes to w. In $\tilde{O}(mn^\delta)$ time compute the distances between all $s \in W$ and $v \in V$. Note that $\delta < 1 - \delta < \omega/3$, and thus so far we have spent $\tilde{O}(mn^\delta + mn^{1-\delta}) \le \tilde{O}(mn^{\omega/3})$ time.

Lemma 4.2. *Either there are two nodes $s_1, s_2 \in S$ that cover the graph with radius $\frac{5R}{3}$, or w.h.p for some $s_3 \in W$ we have $d(s_3, c_i) \le \frac{R}{3}$ for $i = 1$ or $i = 2$.*

Proof. If $\exists s_1, s_2 \in S$ such that $d(s_1, c_1) \leq \frac{2R}{3}$ and $d(s_2, c_2) \leq \frac{2R}{3}$, then any vertex $v \in V$ will have either $d(s_1, v) \leq \frac{5R}{3}$ or $d(s_2, v) \leq \frac{5R}{3}$.

Otherwise, for some c_j, $j = 1, 2$, $d(s, c_j) > \frac{2R}{3}$ for all $s \in S$. As w is picked to be the furthest node from S we have that $d(w, S) > \frac{2R}{3}$. Since S is large enough, w.h.p it hits the closest n^δ nodes to every vertex in the graph. Thus in particular $S \cap W \neq \emptyset$. Let $s \in S \cap W$. Since $d(w, s) > \frac{2R}{3}$ and every node closer to w than s is in W, W must contain all nodes at distance $\leq \frac{2R}{3}$ from w. Next note that for one of $i = 1$ or $i = 2$, $d(w, c_i) \leq R$. Let s_3 be the node on the shortest path between w and c_i at distance exactly $2R/3$ to w, $d(w, s_3) = \frac{2R}{3}, d(s_3, c_i) \leq \frac{R}{3}$. By the above observation, s_3 must be in W and is at distance $\leq R/3$ from c_i, as desired. ∎

We begin by handling the first case, when there exist $s_1, s_2 \in S$ that cover all vertices with radius $\frac{5R}{3}$. Build an $|S|$ by $|V|$ matrix X such that for any $s \in S$ and $v \in V$, $X[s, v] = 0$ if $d(s, v) \leq 5R/3$, and $X[s, v] = 1$ otherwise. Multiply X by X^t. The running time exponent for this product is

$$\omega(1-\delta, 1, 1-\delta) \leq \delta + \omega \cdot (1-\delta) = \omega - \delta(\omega - 1) = \omega - (2\omega - 3)/3 = 1 + \omega/3.$$

By construction, $XX^t[s, s'] = 0$ if and only if s and s' together cover all vertices of V at distance $5R/3$. Therefore $XX^t[s_1, s_2] = 0$, and so we are able to find this pair.

We are left to handle the second case. We can assume that case 1 did not work, and so there is a node $s_1 \in W$, s.t. $d(s_1, c_1) \leq R/3$ (up to relabeling c_1, c_2). Set $\gamma = (\omega^2 - 3\omega + 3)/(3\omega - 3)$ and let T be a random subset of V of size $O(n^{1-\gamma} \log n)$. Note that this choice of γ, δ gives:

$$\gamma + \delta = (\omega^2 - 3\omega + 3 + 2\omega - 3)/(3\omega - 3) = \omega/3,$$

and

$$1 - \gamma \leq (1-\gamma) + \delta(\omega - 2) = \frac{3\omega - 3 - \omega^2 + 3\omega - 3 + 2\omega^2 - 4\omega - 3\omega + 6}{3\omega - 3} = \frac{\omega^2 - \omega}{3\omega - 3} = \omega/3.$$

Run BFS from T to compute all distances between T and V in time $O(mn^{1-\gamma} \log n) \leq \tilde{O}(mn^{\omega/3})$. For every $s \in W$, define $U(s)$ to be the nodes $v \in V$ with $d(s, v) > \frac{4R}{3}$. Note that if $d(s_1, c_1) \leq \frac{R}{3}$, then all the nodes in $U(s_1)$ are at distance $> R$ from c_1, and hence must be covered by c_2 within distance R. Further define $w(s)$ to be the furthest node from T in $U(s)$, and let $W(s)$ be the closest n^γ nodes to $w(s)$. We again show that one of two cases holds.

Lemma 4.3. *Given s_1 such that $d(s_1, c_1) \leq \frac{R}{3}$, w.h.p either there exists a node $s_2 \in W(s_1)$ such that s_1, s_2 cover all vertices with radius $\frac{5R}{3}$, or $Q := \bigcap_{t \in T \cap U(s_1)} B(t, R) \neq \emptyset$ and for any $s_2' \in Q$, the pair s_1, s_2' cover all vertices with radius $\frac{5R}{3}$.*

Proof. If $d(w(s_1), T) > \frac{R}{3}$, then by the same argument as we made in the proof of Lemma 4.2, since T hits $W(s_1)$ w.h.p, all nodes of distance $\leq \frac{R}{3}$ from $w(s_1)$ will be contained in $W(s_1)$. As previously noted, $d(w(s_1), c_2) \leq R$ and so there exists a node s_2 on the shortest path from $w(s_1)$ to c_2 of distance $d(s_2, w(s_1)) = \frac{R}{3}$ and $d(s_2, c_2) \leq \frac{2R}{3}$. Therefore, $s_2 \in W(s_1)$ and the nodes s_1, s_2 cover all vertices with radius $\frac{5R}{3}$.

Otherwise, any node $u \in U(s_1)$ has $d(u, T) \leq \frac{R}{3}$. Since all nodes in $U(s_1)$ are covered by c_2, $c_2 \in B(t, R)$ for all $t \in T \cap U(s_1)$. Therefore, $c_2 \in Q$ so $Q \neq \emptyset$.

Let $s_2' \in Q$ and let $u \in V$. If $d(s_1, u) > \frac{4R}{3}$ then $u \in U(s_1)$ and so there exists a node $t \in T$ such that $d(t, u) \leq \frac{R}{3}$. If $d(s_1, t) \leq \frac{4R}{3}$, then $d(s_1, u) \leq \frac{5R}{3}$. Otherwise, $t \in T \cap U(s_1)$ and so $d(s_2', t) \leq R$, in which case $d(s_2', u) \leq \frac{4R}{3}$. We conclude that s_1, s_2' cover all vertices with radius $\frac{5R}{3}$. ∎

To handle the first case of Lemma 4.3, for every $s_1 \in W$ run BFS from all nodes in $W(s_1)$ in time $\tilde{O}(mn^{\gamma+\delta}) \leq \tilde{O}(mn^{\omega/3})$. For every $s_2 \in W(s_1)$, check if the pair s_1, s_2 cover all vertices with radius $\frac{5R}{3}$, in total time $O(n^{1+\gamma+\delta}) \leq O(n^{1+\omega/3})$. If we are in the first case, for the $s_1 \in W$ such that $d(s_1, c_1) \leq \frac{R}{3}$ we will find the appropriate s_2 and complete the algorithm.

To handle the second case of Lemma 4.3, create an $|W| = \tilde{O}(n^\delta)$ by $|T| = \tilde{O}(n^{1-\gamma})$ Boolean matrix A with rows indexed by W and columns indexed by T. Define

$$A[s, t] = 1 \text{ iff } d(s, t) > 4R/3.$$

Next, create an $\tilde{O}(n^{1-\gamma})$ by n Boolean matrix B with rows indexed by T and columns indexed by V. Define

$$B[t,v] = 1 \text{ iff } d(t,v) > R.$$

Multiply A and B in time $\tilde{O}(n^{\omega(\delta,1-\gamma,1)})$. Notice that by our choice of parameters we get

$$\omega(\delta, 1-\gamma, 1) \le (1-\gamma-\delta) + (1-\delta) + \omega\delta = 2 - \omega/3 + \delta(\omega - 1) = 2 - \omega/3 + (2\omega - 3)/3 = 1 + \omega/3.$$

Consider the s_1-row of AB, where $d(s_1, c_1) \le \frac{R}{3}$. $AB[s_1, v] = 0$ if and only if $v \in Q$. Therefore, after computing AB, for every $s_1 \in W$ we can look for $AB[s_1, s_2'] = 0$ and then check if s_1, s_2' cover all vertices with radius $\frac{5R}{3}$. As all the distances have already been computed, and we only need to check one pair for every $s_1 \in W$, this step takes $O(|W| \cdot n) \le \tilde{O}(n^{1+\omega/3})$.

The total running time (as $m \ge n-1$) is within polylog factors $mn^{\omega/3}$. If $\omega = 2$, the running time is $\tilde{O}(mn^{2/3})$. If $\omega > 2$, one can obtain some improvements using rectangular matrix multiplication. ∎

We note that similar to Subsection 4.4, one can obtain an $(5/3, O(M))-$approximation algorithm running in time $\tilde{O}(mn^{\omega/3})$ for graphs with integer weights bounded by $M \le \text{poly}(n)$.

4.2 3/2 Approximation for k-Center Next, for k-center for any k, we present a simple 3/2 approximation algorithm that incurs an additive error of $+\frac{1}{2}$ when the k-radius is odd.

First we see a lemma we will use in many of our algorithms, which generalized Lemma 4.2 to any value of k. Let G be a graph with k-radius $\le R$ achieved by centers $c_1, c_2, \ldots, c_k$. Let $S \subseteq V$ be a random sample of size $O(n^{1-\delta}\log n)$ and let w be the furthest node in V from S. Take W to be the closest n^δ vertices to the vertex w.

LEMMA 4.4. *For any $r \in \mathbb{N}$, either there exist k nodes $t_1, t_2, \ldots, t_k \in S$ that cover the graph with radius $R + r$, or w.h.p there exists some $s_1 \in W$ such that $d(s_1, c_i) \le R - r$ for some $i \in [k]$, w.l.o.g $i = 1$.*

Proof. If for every center $d(c_i, S) \le r$, denote the closest node to c_i in S by t_i. Now these k nodes $t_1, t_2, \ldots, t_k$ cover the entire graph with radius $R + r$, as any node $v \in V$ has $d(v, c_i) \le R$ for some $i \in [k]$, and thus $d(v, t_i) \le R + r$.

Otherwise, there exists a center such that $d(c_i, S) > r$ and therefore $d(w, S) > r$. W.h.p, the set S hits the closest n^δ nodes to every vertex in the graph, and so S hits W. Therefore, there exists a node $u \in W \cap S$ such that $d(w, u) > r$. By the definition of W, the set must contain all the nodes closer to w than u and so it contains all nodes of distance $\le r$ from w.

For some $i \in [k]$, $d(w, c_i) \le R$. Consider the shortest path between w and c_i. There exists a vertex s_1 on the path that has $d(w, s_1) = r$ and $d(s_1, c_i) \le R - r$. By the above observation, we have that $s_1 \in W$. ∎

Using this lemma we can now prove our 3/2 approximation algorithm.

THEOREM 4.5. *Given an unweighted, undirected graph G, there is an algorithm that computes a $(3/2, 1/2)$-approximation to the k-center w.h.p. and runs in time*

- $\tilde{O}(n^{\omega + 1/(\omega+1)})$ *time for $k = 3$,*
- $\tilde{O}(n^{k-(3-\omega)+1/(k+1)})$ *for $k \ge 4$, and*
- $n^{k-1+1/(k+1)+o(1)}$ *time for $k \ge 10$.*

Proof. For any integer R we will construct an algorithm that finds a set of k points that cover all vertices with radius $\le R + \lceil \frac{R}{2} \rceil$, if $R_k(G) \le R$, thus determining if $R_k(G) > R$ or $R_k(G) \le R + \lceil \frac{R}{2} \rceil$. By performing a binary search over the possible values of R, this algorithm achieves a 3/2 approximation to the k-center if $R_k(G)$ is even. If $R_k(G)$ is odd, we obtain a $(3/2, 1/2)$-approximation.

Assume $R_k(G) \le R$ and begin by computing APSP in $O(n^\omega)$ time via Seidel's algorithm [Sei95]. Sample a vertex set S of size $|S| = O(n^{1-\delta}\log n)$. Let w be the furthest point from S and take W to be the closest n^δ nodes to w. By Lemma 4.4, either there exist k points in S that cover the graph with radius $R + \lceil \frac{R}{2} \rceil$ or there exists a point $s_1 \in S$ such that $d(s_1, c_1) \le R - \lceil \frac{R}{2} \rceil \le \frac{R}{2}$.

To check if we are in the first case, construct an $|S|^{\lceil k/2 \rceil} \times n$ matrix A with rows indexed by $\lceil k/2 \rceil$-tuples of points in S and columns indexed by V, such that $A[(t_1, \ldots, t_{\lceil k/2 \rceil}), v] = 0$ if $\exists i \ d(v, t_i) \le \lceil \frac{3R}{2} \rceil$ and 1 otherwise. Similarly, construct an $n \times |S|^{\lfloor k/2 \rfloor}$ matrix B such that $B[v,(t_1,\ldots,t_{\lfloor k/2\rfloor})] = 0 \iff \exists i\ d(v,t_i) \le \lceil \frac{3R}{2} \rceil$. Multiply A by B. By construction, $AB[(t_1,\ldots,t_{\lceil k/2\rceil}),(t'_1,\ldots,t'_{\lfloor k/2\rfloor})] = 0$ if and only if the points $(t_1,\ldots,t_{\lceil k/2\rceil},t'_1,\ldots,t'_{\lfloor k/2\rfloor})$ cover all vertices of V with radius $\lceil \frac{3R}{2} \rceil$. Thus, if we are in the first case, where there exist $t_1,\ldots,t_k \in S$ that cover V with radius $\lceil \frac{3R}{2} \rceil$, we have $AB[(t_1,\ldots,t_{\lceil k/2\rceil}),(t_{\lceil k/2\rceil+1},\ldots,t_k)] = 0$ we are able to find this k-tuple at this point.

Otherwise, we have a point $s_1 \in W$ such that $d(s_1,c_1) \le \frac{R}{2}$, and hence s_1 covers all nodes with radius $\le \frac{3R}{2}$. We now search for the remaining $k-1$ centers. To do so we construct an $n^{\delta+\lceil k/2\rceil-1} \times n$ matrix C with rows indexed by $W \times V^{\lceil k/2\rceil-1}$ and columns indexed by V where $C[(s_1,v_2,\ldots,v_{\lceil k/2\rceil}),v] = 0$ if $d(v,\{s_1,v_2,\ldots,v_{\lceil k/2\rceil}\}) \le \frac{3R}{2}$ and 1 otherwise. Similarly define an $n \times n^{\lfloor k/2\rfloor}$ matrix D where $D[v,(v_1,\ldots,v_{\lfloor k/2\rfloor})] = 0 \iff \exists i\ d(v,v_i) \le \frac{3R}{2}$. Multiply C by D and note that, as before, $CD[(s_1,v_2,\ldots,v_{\lceil k/2\rceil}),(v_{\lceil k/2\rceil+1},\ldots,v_k)] = 0$ if and only if the points $s_1,v_2,\ldots,v_k$ cover all vertices with radius $\le \frac{3R}{2}$. Therefore, we are able to find such a k-tuple.

The runtime of this algorithm is dominated by computing APSP and the two matrix products. Assuming $k \ge 3$, we can bound the final runtime within polylogs by

$$n^\omega + \mathrm{MM}\left(n^{(1-\delta)\lceil k/2\rceil}, n, n^{(1-\delta)\lfloor k/2\rfloor}\right) + \mathrm{MM}\left(n^{\delta+\lceil k/2\rceil-1}, n, n^{\lfloor k/2\rfloor}\right).$$

Let's consider the running time for $k = 3$. We bound all rectangular matrix multiplication bounds by appropriate square matrix multiplications; we also set $\delta = \frac{1}{\omega+1}$:

$$\begin{aligned}
&\tilde{O}\left(n^\omega + \mathrm{MM}\left(n^{2(1-\delta)}, n, n^{1-\delta}\right) + \mathrm{MM}\left(n^{\delta+1}, n, n\right)\right) \le \\
&\tilde{O}\left(n \cdot \mathrm{MM}\left(n^{1-\delta}, n^{1-\delta}, n^{1-\delta}\right) + n^\delta \cdot \mathrm{MM}\left(n,n,n\right)\right) \le \\
&\tilde{O}\left(n^{1+\omega(1-\delta)} + n^{\omega+\delta}\right) = \tilde{O}(n^{\omega+\frac{1}{\omega+1}}).
\end{aligned}$$

We proceed similarly for $k \ge 4$ but setting $\delta = \frac{1}{1+k}$. We omit the $\tilde{O}$ below:

$$\begin{aligned}
&n^\omega + \mathrm{MM}\left(n^{(1-\delta)\lceil k/2\rceil}, n, n^{(1-\delta)\lfloor k/2\rfloor}\right) + \mathrm{MM}\left(n^{\delta+\lceil k/2\rceil-1}, n, n^{\lfloor k/2\rfloor}\right) \le \\
&n^\omega + n^{(1-\delta)\lceil k/2\rceil-1} \cdot n^{(1-\delta)\lfloor k/2\rfloor-1} \cdot \mathrm{MM}(n,n,n) + n^{\delta+\lceil k/2\rceil-2} \cdot n^{\lfloor k/2\rfloor-1} \cdot \mathrm{MM}(n,n,n) = \\
&n^\omega + n^{(1-\delta)k+\omega-2} + n^{\delta+k+\omega-3} \le n^{k+\frac{1}{k+1}+\omega-3}.
\end{aligned}$$

If $\omega > 2$ we can improve on this runtime using the current fastest techniques for rectangular matrix multiplication. We give the best improvement which holds for $k \ge 10$. We again set $\delta = \frac{1}{1+k}$. For $k \ge 10$ we can use the fact that for all $a < 0.3213$, $\mathrm{MM}(N, N^a, N) \le N^{2+o(1)}$ (see [VXXZ24]) to bound the running time. We omit $n^{o(1)}$ factors below:

$$\begin{aligned}
&n^\omega + \mathrm{MM}\left(n^{(1-\delta)\lceil k/2\rceil}, n, n^{(1-\delta)\lfloor k/2\rfloor}\right) + \mathrm{MM}\left(n^{\delta+\lceil k/2\rceil-1}, n, n^{\lfloor k/2\rfloor}\right) \le \\
&n^\omega + n^{(1-\delta)(\lceil k/2\rceil-\lfloor k/2\rfloor)} \cdot \mathrm{MM}\left(n^{(1-\delta)\lfloor k/2\rfloor}, n, n^{(1-\delta)\lfloor k/2\rfloor}\right) + n^{1-\delta+\lceil k/2\rceil-\lfloor k/2\rfloor} \cdot \mathrm{MM}\left(n^{\delta+\lfloor k/2\rfloor-1}, n, n^{\delta+\lfloor k/2\rfloor-1}\right) \le \\
&n^\omega + n^{(1-\delta)k} + n^{\delta-1+k} = n^{k-1+\frac{1}{k+1}},
\end{aligned}$$

where the last inequality holds because for $k \ge 10$, $0.3213(1-\delta)\lfloor k/2\rfloor \ge 0.3213(1-1/(k+1))5 > 1.5(1-1/11) > 1$ and $0.3213(\delta + \lfloor k/2\rfloor - 1) \ge 0.3213 \cdot 4 > 1$. □

4.3 $2-\frac{1}{2k-1}$ Approximation for k-Center

Next we show a faster algorithm for approximating k-center, with the approximation factor growing with k. Our algorithm will begin the same way as the 3/2 approximation, by sampling a set S that either contains k points that can act as approximate centers, or finding a set W which has a point $s_1 \in W$ such that $d(s_1,c_1)$ is small. We then show that we can find $k-1$ centers in S to cover all vertices along with s_1, or we find a point s_2 which is close to c_2. We repeat this process until we have $s_1,s_2,\ldots,s_k$ such that $d(s_i,c_i)$ is bounded for every i, and use these k points as approximate centers. We first prove a combinatorial version of our algorithm, with a simple running time analysis. In the following section we show how to improve the runtime using fast matrix multiplication.

THEOREM 4.6. *There exists an algorithm running in $\tilde{O}(mn+n^{k/2+1})$ time that, when given an integer R and an unweighted, undirected graph G, determines w.h.p one of the following*

1. $R_k(G) > R$,
2. $R_k(G) \leq 2R - \left\lfloor \frac{R}{2k-1} \right\rfloor$.

By binary searching over the possible values of R we obtain the desired approximation. If $R_k(G)$ is divisible by $(2k-1)$, this algorithm gives a multiplicative $(2-\frac{1}{2k-1})$ approximation. Otherwise, we incur an additive error of at most $+\left(1-\frac{1}{2k-1}\right)$. This gives us our desired result:

THEOREM 4.7. *There exists an algorithm running in $\tilde{O}(mn+n^{k/2+1})$ that computes a $\left(2-\frac{1}{2k-1}, 1-\frac{1}{2k-1}\right)$-approximation to k-center.*

Proof of Theorem 4.6. Denote by $\alpha = \left\lfloor \frac{R}{2k-1} \right\rfloor$. Assume $R_k(G) \leq R$, we will show in this case that we are able to find a set of k points that cover all vertices with radius $2R-\alpha$. If we are unable to do so we determine that $R_k(G) > R$. As with the 3/2 approximation, we begin by computing APSP in time $O(mn)$ and then randomly sampling a set S of size $|S| = O(\sqrt{n}\log n)$. Let w_1 be the furthest node from S and let W_1 be the closest $\sqrt{n}$ vertices to w_1.

By Lemma 4.4, either there exist k points in S that cover all vertices with radius $2R-\alpha$, or there exists a point $s_1 \in W_1$ with $d(s_1,c_1) \leq \alpha$. To check if we are in the first case, check for every every k-tuple in S^k whether it covers all vertices with radius $2R-\alpha$, doing so takes $n \cdot |S|^k = \tilde{O}(n^{k/2+1})$ time.

If we have not finished, we can assume $\exists s_1 \in W_1\ d(s_1,c_1) \leq \alpha$. Fix a vertex $s_1 \in W_1$ and define $U = B(s_1, R+\alpha)^c$, the points of distance $> R+\alpha$ from s_1, and $Y = B(s_1, 2R-\alpha)^c$. If we have the correct s_1 (i.e. $d(s_1,c_1) \leq \alpha$), then none of the points in U are covered by c_1. Let w_2 be the furthest point in U from S. In particular, there exists a center c_i for $i \neq 1$ such that $d(c_i, w_2) \leq R$, w.l.o.g $i=2$. Let W_2 be the $\sqrt{n}$ closest nodes to w_2. If $d(w_2,S) > R-3\alpha$, then W_2 contains all the vertices of distance $\leq R-3\alpha$ from w_2, and therefore, there exists a point $s_2 \in W_2$ on the shortest path from w_2 to c_2 such that $d(w_2,s_2) = R-3\alpha$ and $d(s_2,c_2) \leq 3\alpha$. Otherwise, we claim that $k-1$ points in S can cover Y with radius $2R-\alpha$. More generally, we show the following key lemma.

LEMMA 4.8. *Let $1 \leq i < k$, let $r, \alpha \in \mathbb{Z}$ be such that $r+\alpha \leq R$, and let $C_i = \{s_1, \ldots, s_i\}$ be a set of i points where $d(s_j, c_j) \leq r$ for every $1 \leq j \leq i$. Define $U = B(C_i, R+r)^c, Y = B(C_i, 2R-\alpha)^c$. Let w_{i+1} be the furthest point in U from S and let W_{i+1} be the closest $\sqrt{n}$ nodes to w_{i+1}. One of the following two holds.*

1. *$\exists s_{i+1} \in W_{i+1}$ such that $d(s_{i+1}, c_j) \leq r+2\alpha$ for some $j > i$, w.l.o.g $j = i+1$.*
2. *There exist $k-i$ points in S that cover Y with radius $2R-\alpha$.*

Proof. First consider the case when $d(w_{i+1}, S) > R-r-2\alpha$. By the argument we have shown before, since S hits W_{i+1} and w_{i+1} is covered by a center c_j for some $j > i$, there exists a point $s_{i+1} \in W_{i+1}$ on the shortest path from w_{i+1} to c_j that has $d(w_{i+1}, s_{i+1}) = R-r-2\alpha$ and $d(s_{i+1}, c_j) \leq r+2\alpha$.

Otherwise, any point $u \in U$ has $d(u,S) \leq R-r-2\alpha$. We claim that in this case, any center c_j that has $d(c_j, y) \leq R$ for some point $y \in Y$, must have $d(c_j, S) \leq R-\alpha$.

If $c_j \in U$, then $d(c_j,S) \leq R-r-2\alpha \leq R-\alpha$ and we are done. Otherwise, fix a shortest path from c_j to y, let x be the first vertex on the path that belongs to U (it must exist since $y \in Y \subseteq U$). Since $x \in U$ we know $d(x,S) \leq R-r-2\alpha$. Thus, it suffices to show that $d(c_j,x) \leq \alpha + r$, in which case we would have

$$d(c_j,S) \leq d(c_j,x) + d(x,S) \leq \alpha + r + R - r - 2\alpha = R - \alpha.$$

Let p be the predecessor of x on the shortest path from c_j to y, see Fig. 6. By our choice of x we know $p \notin U$ and so $d(p, s_\ell) \leq R+r$ for some $1 \leq \ell \leq i$, by the definition of U. Then, since $d(s_\ell, y) > 2R-\alpha$, we have by the triangle inequality,

$$d(p,y) \geq d(s_\ell,y) - d(p,s_\ell) > 2R - \alpha - (R+r) = R - r - \alpha.$$

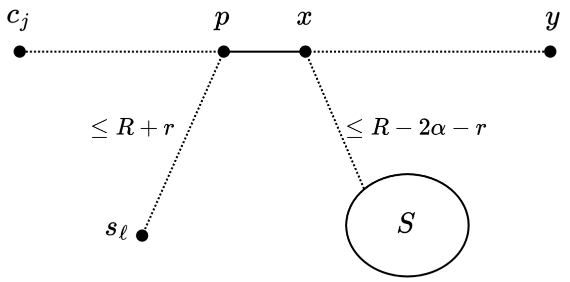

Figure 6: Shortest Path from c_j to y

Now, since $d(c_j, y) \leq R$ we conclude that $d(c_j, p) < r + \alpha$ and so $d(c_j, x) < r + \alpha + 1$. Since $d(c_j, x), r$ and α are all integers we can conclude that $d(c_j, x) \leq r + \alpha$.

Therefore, since the points in Y are covered by $k - i$ centers $c_{i+1}, \ldots, c_k$, there exist $k - i$ points in S, each within distance $R - \alpha$ of the appropriate center, that cover Y with radius $2R - \alpha$. ◻

Using this lemma, we perform $k - 1$ iterations of the following algorithm. At step $i + 1$ we have defined $W_1, \ldots, W_i$ and are guaranteed to have $s_1 \in W_1, \ldots, s_i \in W_i$ such that $d(s_j, c_j) \leq (2j - 1)\alpha$ for every $j \leq i$. We check if there exists a k-tuple in $W_1 \times W_2 \times \ldots \times W_i \times S^{k-i}$ that covers all vertices in V with radius $2R - \alpha$ and if so, return this k-tuple. This takes us $\tilde{O}(n^{k/2+1})$ time. Otherwise, for every i-tuple in $W_1 \times \ldots \times W_i$ define w_{i+1} and W_{i+1} as in Lemma 4.8. By Lemma 4.8 we know that there exists $s_{i+1} \in W_{i+1}$ such that $d(s_{i+1}, c_{i+1}) \leq (2i + 1)\alpha$ and so we can begin step $i + 1$.

If we were to perform this step k times we would have that $d(s_k, c_k) \leq (2k - 1)\alpha$. In order for $s_1, \ldots, s_k$ to cover all vertices in V with radius $2R - \alpha$, we would need $\alpha \leq \frac{R}{2k}$, which would give us a $2 - \frac{1}{2k}$ approximation. Instead, we do something slightly different at the last step that allows us to get a $2 - \frac{1}{2k-1}$ approximation, using the approach from our 5/3 approximation for 2-center.

At step $k - 1$, we have $W_1, \ldots, W_{k-1}$ such that $\exists s_i \in W_i \; d(s_i, c_i) \leq (2k - 3)\alpha$ for any $i < k$. For every $(k - 1)$-tuple $\tau \in W_1 \times \ldots \times W_{k-1}$ define $U^\tau, Y^\tau, w_k^\tau, W_k^\tau$ as in Lemma 4.8, using $r = (2k - 3)\alpha$. We use the following lemma, generalizing Lemma 4.3.

Lemma 4.9. *Given $\tau = (s_1, \ldots, s_{k-1})$ such that $d(s_i, c_i) \leq (2k - 3)\alpha$ for every $i < k$, one of the following two holds.*

1. *$\exists s_k \in W_k^\tau$ such that $d(s_k, c_k) \leq (2k - 2)\alpha$.*
2. *$Q \coloneqq \bigcap_{t \in S \cap U^\tau} B(t, R) \neq \emptyset$, and any point $s'_k \in Q$ covers all points in Y^τ with radius $2R - \alpha$.*

We will prove a slightly more general statement, which we will use in other contexts later on in the paper. Setting $r = (2k - 3)\alpha, \delta = \frac{1}{2}$ in the following lemma gives us Lemma 4.9.

Lemma 4.10. *Let $r, \alpha \in \mathbb{Z}$ be such that $r + \alpha \leq R$, let S be a random subset of vertices of size $O(n^{1-\delta} \log n)$ and $C = \{s_1, \ldots, s_{k-1}\}$ be nodes such that $d(s_i, c_i) \leq r$ for every $i < k$. Define $U = B(C, R + r)^c$ and $Y = B(C, 2R - \alpha)^c$. Let w be the furthest point in U from S and let W be the closest n^δ vertices to w. One of the following two holds.*

1. *$\exists s_k \in W$ such that $d(s_k, c_k) \leq r + \alpha$.*
2. *$Q \coloneqq \bigcap_{t \in S \cap U} B(t, R) \neq \emptyset$, and any point $s'_k \in Q$ covers all points in Y with radius $2R - \alpha$.*

Proof. If $d(w, S) > R - (r + \alpha)$, then by the argument we have seen in previous proofs we have that $\exists s_k \in W$ such that $d(s_k, c_k) \leq r + \alpha$. Otherwise, any point $u \in U$ has $d(u, S) \leq R - (r + \alpha)$.

Any point $t \in S \cap U$ is not covered by $c_1, \ldots, c_{k-1}$ and must be covered by c_k. Thus, $c_k \in B(t, R)$ for any $t \in S \cap U$ and $c_k \in Q$, meaning $Q \neq \emptyset$.

Let s'_k be an arbitrary vertex in Q and let $y \in Y$. Let $t \in S$ be the closest vertex to y in S. Since $Y \subseteq U$, $d(t, y) \leq R - (r + \alpha)$. If $t \notin U$, then there exists some s_i such that $d(s_i, t) \leq R + r$ and hence $d(s_i, y) \leq R + r + R - (r + \alpha) = 2R - \alpha$. However, this would imply $y \notin Y$ and so we conclude $t \in U$. Therefore, by definition of Q, $d(s'_k, t) \leq R$, and so $d(s'_k, y) \leq R + R - (r + \alpha) \leq 2R - \alpha$. ∎

We can now finish the algorithm. For every $\tau \in W_1 \times \ldots \times W_{k-1}$, compute W_k^τ and check if any point in $s_k \in W_k^\tau$ covers all vertices in Y^τ within distance $R + (2k-2)\alpha \leq 2R - \alpha$. If so, return (τ, s_k). Otherwise, compute Q, and if it is not empty check if an arbitrary point in $s'_k \in Q$ covers all vertices in Y^τ within distance $2R - \alpha$. If so, return (τ, s'_k). For some $\tau \in W_1 \times \ldots \times W_{k-1}$ we know that the conditions of Lemma 4.9 hold, and so we will be able to find the appropriate s_k or s'_k and complete the algorithm.

We describe the entire algorithm in Algorithm 2.

Algorithm 2 $\left(2 - \frac{1}{2k-1}\right)$-Approximation Algorithm for k-Center

```
1: Input: $G, R, k$ such that $R_k(G) \leq R$.
2: Output: Vertices $s_1, s_2, \ldots, s_k$ that cover all $V$ with radius $\leq 2R - \alpha$, where $\alpha := \left\lfloor \frac{R}{2k-1} \right\rfloor$
3: Compute APSP, randomly sample a subset $S \subset V$ of size $|S| = O(\sqrt{n} \log n)$.
4: Step$(R, k, S, 0, ())$.
5:
6: procedure Step$(R, k, S, i, (s_1, \ldots, s_i))$
7:     $w_{i+1} \leftarrow$ furthest vertex from $S$ in $B(\{s_1, s_2, \ldots, s_i\}, R + (2i-1)\alpha)^c$.
8:     $W_{i+1} \leftarrow$ closest $\sqrt{n}$ vertices to $w_{i+1}$.
9:     if $i < k - 1$ then
10:        if $\exists (t_{i+1}, \ldots, t_k) \in S^{k-i}$ that covers $B(\{s_1, s_2, \ldots, s_i\}, 2R - \alpha)^c$ with radius $2R - \alpha$ then
11:            return $s_1, \ldots, s_i, t_{i+1}, \ldots, t_k$.
12:        end if
13:        for $s_{i+1} \in W_{i+1}$ do
14:            Step$(R, k, S, i + 1, (s_1, \ldots, s_i, s_{i+1}))$.
15:        end for
16:    else                                                    ▷ Last step, $i = k - 1$
17:        if $\exists s_k \in W_k$ that covers $B(\{s_1, s_2, \ldots, s_{k-1}\}, 2R - \alpha)^c$ with radius $2R - \alpha$ then
18:            return $s_1, \ldots, s_{k-1}, s_k$.
19:        end if
20:        $Q \leftarrow \bigcap_{t \in S \cap B(\{s_1, \ldots, s_{k-1}\}, R + (2k-3)\alpha)^c} B(t, R)$, $s'_k \leftarrow$ arbitrary point in $Q$.
21:        if $(s_1, \ldots, s_{k-1}, s'_k)$ cover all vertices with radius $\leq 2R - \alpha$ then
22:            return $(s_1, \ldots, s_{k-1}, s'_k)$.
23:        end if
24:    end if
25: end procedure
```

Runtime: Denote by T_i the runtime of the function Step$(R, k, S, i, (s_1, \ldots, s_i))$. The final runtime of Algorithm 2 is $O(mn) + T_0 = \tilde{O}(mn + n^{k/2+1})$, as we prove in the following claim.

Claim 4.11. $T_i = \tilde{O}(n^{(k-i)/2+1})$.

Proof. For $i < k - 1$, the function Step$(R, k, S, i, (s_1, \ldots, s_i))$ checks for $|S|^{k-i}$ tuples whether they cover a set of points, taking $O(n^{(k-i)/2+1})$, and then runs Step$(R, k, S, i + 1, (s_1, \ldots, s_{i+1}))$ on $|W_{i+1}| = \sqrt{n}$ different values of s_{i+1}. Therefore,

$$T_i = \tilde{O}(n^{(k-i)/2+1}) + \sqrt{n} \cdot T_{i+1}.$$

We prove the claim by induction on i. For $i = k-1$, Step$(R, k, k-1, S, (s_1, \ldots, s_{k-1}))$ checks for $\sqrt{n}$ vertices if they cover a set of point, and then computes the intersection of $\leq |S| = \tilde{O}(\sqrt{n})$ balls and checks if a single k-tuple covers all points with a sufficiently small radius. Both these steps run in $\tilde{O}(n^{3/2})$. Now assume the claim is true for $i+1$, we have

$$\begin{aligned} T_i &= \tilde{O}(n^{(k-i)/2+1}) + \sqrt{n} \cdot T_{i+1} \\ &= \tilde{O}(n^{(k-i)/2+1}) + \sqrt{n} \cdot \tilde{O}(n^{(k-i-1)/2+1}) \\ &= \tilde{O}(n^{(k-i)/2+1}). \end{aligned}$$

□

□

4.3.1 Improving the Runtime Using Fast Matrix Multiplication In this section we use fast matrix multiplication to speed up the running time of the algorithm presented in Theorem 4.7. To introduce the techniques, we first assume in our analysis that $\omega = 2$. We then get rid of that assumption and prove Theorem 1.4.

The approach will be analogous to the algorithm above, but we will produce a new random sample S_i during each stage of the algorithm, instead of using the same set S in all stages. The sample sizes $|S_i|$ and the sizes of the sets W_i will vary (unlike before when they were $\sqrt{n}$). We then use fast matrix multiplication to check if there exists a set of points covering another set with sufficiently small radius.

We first prove the following result, which generalizes our earlier algorithm for $k = 2$ in Theorem 4.1. We still include the result of Theorem 4.1 as a special case of the following theorem.

Theorem 4.12. *If $\omega = 2$, there is an algorithm that computes w.h.p. a $\left(2 - \frac{1}{2k-1}, 1 - \frac{1}{2k-1}\right)$-approximation to the k-center of any unweighted, undirected graph, and runs in time*

- *$\tilde{O}(mn^{2/3})$ for $k = 2$, and*
- *$\tilde{O}(n^{k/2+22/9(k+1)})$ for $k \geq 3$.*

Proof. We begin by recursively defining a sequence of values $\delta_{k-1}, \delta_{k-2}, \ldots, \delta_0$. For every $1 \leq i \leq k$, let $t_{k-i} = 1 + \sum_{j=1}^{i} \delta_{k-j}$.

- $\delta_{k-1} = \delta_{k-2} = \frac{1}{3}$,
- $\delta_{k-3} = \frac{4}{9}$, and
- For $4 \leq i \leq k$, $\delta_{k-i} = \frac{i - t_{k-i+1}}{i+1}$.

We note several properties.

Observation 4.13. *For all $i \geq 4$, $i \cdot (1 - \delta_{k-i}) = \delta_{k-i} + t_{k-i+1} = t_{k-i}$.*

Claim 4.14. *For all $i \geq 3$, $t_{k-i} = \frac{i}{2} + \frac{22}{9(i+1)}$.*

We will prove a slightly more general claim.

Claim 4.15. *Let $\delta_0, \ldots, \delta_j, t_0, \ldots, t_{k-j}$ be such that $t_{k-i} = \delta_{k-i} + t_{k-i+1}$ and $\delta_{k-i} = \frac{i - t_{k-i+1} + c}{i+1}$ for some constant c for all $i > j$. Then there exists a constant a such that $t_{k-i} = \frac{i}{2} + \frac{a}{i+1} + c$ for all $i \geq j$.*

Proof. By induction on i. For $i = j$ pick a such that $t_{k-j} = \frac{j}{2} + \frac{a}{j+1} + c$. Now assume the claim holds for $i-1$.

$$\begin{aligned} t_{k-i} &= t_{k-i+1} + \delta_{k-i} = t_{k-i+1} + \frac{i - t_{k-i+1} + c}{i+1} \\ &= \frac{i-1}{2} + \frac{a}{i} + c + \frac{i - (\frac{i-1}{2} + \frac{a}{i} + c) - c}{i+1} = \\ &= \frac{i-1}{2} + \frac{a}{i} + c + \frac{\frac{i+1}{2} - \frac{a}{i}}{i+1} \\ &= \frac{i-1}{2} + \frac{1}{2} + \frac{a}{i} - \frac{a}{i(i+1)} + c = \frac{i}{2} + \frac{a}{i+1} + c. \end{aligned}$$

□

Proof of Claim 4.14. For $i = 3$, $t_{k-3} = 1 + \frac{1}{3} + \frac{1}{3} + \frac{4}{9} = \frac{19}{9} = \frac{3}{2} + \frac{22}{9 \cdot 4}$. Therefore, by Claim 4.15, for all $i \geq 3$, $t_{k-i} = \frac{i}{2} + \frac{22}{9(k+1)}$. ∎

Claim 4.16. *Let $\delta_0, \ldots, \delta_j, t_0, \ldots, t_{k-j}$ be such that $t_{k-i} = \delta_{k-i} + t_{k-i+1}$ and $t_{k-i} = \frac{i}{2} + \frac{a}{i+1}$ for $a \geq 0$ for all $i > j$. Then $\delta_{k-i} \leq \frac{1}{2}$ for all $i > j$.*

Proof. $\delta_{k-i} = t_{k-i} - t_{k-i+1} = \frac{i}{2} + \frac{a}{i+1} - \frac{i-1}{2} - \frac{a}{i} = \frac{1}{2} - \frac{a}{i(i+1)} \leq \frac{1}{2}$. ∎

We conclude that $\delta_{k-i} \leq \frac{1}{2}$ for all $1 \leq i \leq k$.

Given these values, we can now adjust Algorithm 2. For $k > 2$, we begin by running APSP in $\tilde{O}(n^{\omega}) = \tilde{O}(n^2)$ time [Sei95]. For $k = 2$ we avoid running APSP and instead run BFS from the necessarily points and sets throughout the algorithm. We adjust the subroutine $\textsc{Step}(R, k, S, i, (s_1, \ldots, s_i))$ and optimize the final step of the algorithm in a similar way to Theorem 4.1. We show that the runtime of the resulting algorithm, when $\omega = 2$, is $\tilde{O}(n^{k/2+22/9(k+1)})$.

First, instead of using the set S, the routine $\textsc{Step}(R, k, S, i, (s_1, \ldots, s_i))$ samples a set S_i of size $|S_i| = O(n^{1-\delta_i} \log n)$ and uses this set in place of the set S. We will therefore omit the S from calls to $\textsc{Step}()$ in the remainder of this section. We then define W_{i+1} to be the n^{δ_i} closest vertices to w_{i+1}. W.h.p, the set S_i hits W_{i+1}, so the resulting arguments still hold.

Second, to check whether there exist $(t_{i+1}, \ldots, t_k) \in S_i^{k-i}$ that cover $Z := B(\{s_1, \ldots, s_i\}, 2R - \alpha)^c$ (line 10 in Algorithm 2) we use fast matrix multiplication. Construct a $n^{(1-\delta_i) \cdot \lfloor (k-i)/2 \rfloor} \times |Z|$ boolean matrix X whose rows are indexed by $\lfloor (k-i)/2 \rfloor$-tuples of vertices of S_i and whose columns are indexed by Z. For a tuple τ and a vertex u, $X[\tau, u] = 1$ if $d(\tau, u) > 2R - \alpha$, and $X[\tau, u] = 0$ otherwise. Similarly, we form a $|Z|$ by $\tilde{O}(n^{(1-\delta_i) \cdot \lceil (k-i)/2 \rceil})$ matrix Y whose rows are indexed by Z and whose columns are indexed by $\lceil (k-i)/2 \rceil$-tuples of nodes of S_i. For a tuple τ and a vertex u, $Y[u, \tau] = 1$ if $d(\tau, u) > 2R - \alpha$, and $Y[\tau, u] = 0$ otherwise.

After multiplying X and Y we can determine if a $(k-i)$-tuple of vertices of S_i covers Z within distance $\leq 2R - \alpha$ by finding a 0 entry in XY. By Claim 4.16, $1 - \delta_i \geq \frac{1}{2}$. Therefore, the running time, when $\omega = 2$ and $k - i \geq 4$, is up to polylogarithmic factors

$$\leq \mathrm{MM}(n^{(1-\delta_i) \cdot \lfloor (k-i)/2 \rfloor}, n, n^{(1-\delta_i) \cdot \lceil (k-i)/2 \rceil}) = n^{(1-\delta_i)(k-i)}.$$

If we haven't found an approximate k-center at this point (for $i < k-1$), we run $\textsc{Step}(R, k, i+1, (s_1, \ldots, s_{i+1}))$ for $|W_{i+1}| = n^{\delta_i}$ different values of s_{i+1}. Thus, if we denote by $\hat{T}_i$ the running time of the subroutine $\textsc{Step}$ at stage i, we have for $i < k - 1$,

$$\hat{T}_i = \tilde{O}(\mathrm{MM}(n^{(1-\delta_i) \cdot \lfloor (k-i)/2 \rfloor}, n, n^{(1-\delta_i) \cdot \lceil (k-i)/2 \rceil}) + n^{\delta_i} \cdot \hat{T}_{i+1}).$$

If $i \leq k - 4$ we have,

$$\hat{T}_i = \tilde{O}(n^{(1-\delta_i)(k-i)} + n^{\delta_i} \cdot \hat{T}_{i+1}).$$

Finally, we change the final steps of the algorithm for $i = k - 1$ and $i = k - 2$ using the technique from Theorem 4.1. When calling $\textsc{Step}(R, k, k-1, (s_1, \ldots, s_{k-1}))$ we only check for a point $s_k \in W_k$ that completes the center and do nothing, i.e. we run line 17 but not lines 20-23. Thus, $\hat{T}_{k-1} = \tilde{O}(n^{1+\delta_{k-1}}) = \tilde{O}(n^{4/3})$.

When calling $\textsc{Step}(R, k, k-2, (s_1, \ldots, s_{k-2}))$ we add an additional step. Construct a $|W_{k-1}| \times |S_{k-1}|$ boolean matrix A where $A[s_{k-1}, t] = 1 \iff d(s_{k-1}, t) > R + (2k-3)\alpha$ and a $|S_{k-1}| \times V$ boolean matrix B where $B[t, v] = 1 \iff d(t, v) > R$. Multiply them in time $n^{\omega(\delta_{k-2}, 1-\delta_{k-1}, 1)}$. Now, for every $s_{k-1} \in W_{k-1}$, if there is a zero in the row corresponding to it $AB[s_{k-1}, u] = 0$, check if u covers $B(\{s_1, \ldots, s_{k-1}\}, 2R - \alpha)^c$ with radius $2R - \alpha$. Otherwise, for every $s_{k-1} \in W_{k-1}$ run $\textsc{Step}(R, k, k-1, (s_1, \ldots, s_{k-1}))$. As we showed in the proof of Theorem 4.1, this obtains the same result as our original version of Algorithm Algorithm 2.

The new running time for $\textsc{Step}(R, k, k-2, (s_1, \ldots, s_{k-2}))$ now includes two matrix products and a recursive call to $\textsc{Step}(R, k, k-1, (s_1, \ldots, s_{k-1}))$. So we have, omitting polylogarithmic factors,

$$\begin{aligned}\hat{T}_{k-2} &= n^{\omega(1-\delta_{k-2}, 1, 1-\delta_{k-2})} + n^{\omega(\delta_{k-2}, 1-\delta_{k-1}, 1)} + n^{\delta_{k-2}} \hat{T}_{k-1} \\ &= n^{2-\delta_{k-2}} + n^{2-\delta_{k-1}} + n^{1+\delta_{k-1}+\delta_{k-2}} = n^{5/3}.\end{aligned}$$

Consider the running time for $k = 2$. In this case, $\hat{T}_0 = \hat{T}_{k-2} = n^{5/3}$. In addition to running $\textsc{Step}(R, 2, 0, ())$ in $\hat{T}_0$ time, for $k = 2$ we need to run BFS from all points in the sets S_0, S_1 in time $\tilde{O}(mn^{1-\delta_0} + mn^{1-\delta_1}) \leq \tilde{O}(mn^{2/3})$. Additionally, we run BFS from all points in W_1 in time $O(mn^{\delta_0}) \leq \tilde{O}(mn^{2/3})$ during our call to $\textsc{Step}(R, 2, 0, ())$, and from all points in W_2 in every call to $\textsc{Step}(R, 2, 1, (s_1))$. This takes $O(mn^{\delta_1})$ and is called n^{δ_0} times, so in total runs in time $O(mn^{\delta_0+\delta_1}) \leq \tilde{O}(mn^{2/3})$. Therefore, for $k = 2$ our algorithm runs in time $\tilde{O}(mn^{2/3})$.

We can now use the properties we proved about $\delta_0, \ldots, \delta_{k-1}$ to compute the runtime of our adjusted algorithm for $k > 2$.

Claim 4.17. *For $1 \leq i \leq k$, $\hat{T}_{k-i} = \tilde{O}(n^{t_{k-i}})$.*

Proof. For $i = 1, 2$ we showed $\hat{T}_{k-1} = \tilde{O}(n^{4/3}) = \tilde{O}(n^{t_{k-1}})$ and $\hat{T}_{k-2} = \tilde{O}(n^{5/3}) = \tilde{O}(n^{t_{k-2}})$.

For $i = 3$, since $1/2 > \delta_{k-3} > 0$ we have $(1 - \delta_{k-3}) < 1 < 2(1 - \delta_{k-3})$ and so the matrix product step running time,

$$\mathrm{MM}(n^{(1-\delta_{k-3})}, n, n^{2(1-\delta_{k-3})}) = n^{3-2\delta_{k-3}}.$$

Thus, within polylogarithmic factors

$$\hat{T}_{k-3} = n^{3-2\delta_{k-3}} + n^{\delta_{k-3}} \cdot \hat{T}_{k-2} = n^{3-2\delta_{k-3}} + n^{\delta_{k-3}+t_{k-2}} = n^{19/9} = n^{t_{k-3}}.$$

For $i \geq 4$, assume the claim holds for $i - 1$, then by Observation 4.13,

$$\hat{T}_{k-i} = \tilde{O}(n^{(1-\delta_{k-i})i} + n^{\delta_{k-i}} \cdot \hat{T}_{k-i+1}) = \tilde{O}(n^{(1-\delta_{k-i})i} + n^{\delta_{k-i}+t_{k-i+1}}) = \tilde{O}(n^{t_{k-i}}).$$

□

The final runtime of the algorithm is dominated by running APSP in $\tilde{O}(n^\omega)$ time (and we assumed $\omega = 2$) and calling $\textsc{Step}(R, k, 0, ())$. Therefore, when $k \geq 3$, our algorithm runs in time $\tilde{O}(n^2) + \hat{T}_0 = \tilde{O}(n^2) + \tilde{O}(n^{t_0}) = \tilde{O}(n^{k/2+22/9(k+1)})$. □

We can now compute the runtime of our adjusted Algorithm 2 without the assumption that $\omega = 2$.

Theorem 4.18. *There is an algorithm that computes w.h.p. a $\left(2 - \frac{1}{2k-1}, 1 - \frac{1}{2k-1}\right)$-approximation to the k-center of any unweighted, undirected graph, and runs in time*

- $\tilde{O}(mn^{\omega/3})$ *for $k = 2$,*
- $\tilde{O}(n^{\frac{k}{2}+\frac{\beta_0}{k+1}+(\omega-2)})$ *for $3 \leq k \leq 13$, where $\beta_0 := \frac{-8\omega^2+18\omega+18}{3\omega+3} \approx 1.5516$, and*
- $\tilde{O}(n^{\frac{k}{2}+\frac{\beta_1}{k+1}+o(1)})$ *for $k \geq 13$, where $\beta_1 := \frac{34\omega^2-24\omega-66}{3\omega+3} \approx 6.7533$.*

Proof. We begin by defining slightly different values of $\delta_0, \ldots, \delta_{k-1}$. Let $t_{k-i} = 1 + \sum_{j=1}^{i} \delta_{k-j}$ as before.

- $\delta_{k-1} = \frac{2\omega-3}{3\omega-3}$,
- $\delta_{k-2} = \frac{\omega^2-3\omega+3}{3\omega-3}$, and
- $\delta_{k-3} = \frac{2\omega}{3\omega+3}$.
- For $4 \leq i \leq 13$, let $\delta_{k-i} = \frac{i-t_{k-i+1}+(\omega-2)}{i+1}$.
- For $i \geq 14$, let $\delta_{k-i} = \frac{i-t_{k-i+1}}{i+1}$.

We will again show that $\hat{T}_i = \tilde{O}(n^{t_i})$ and use the following two claims to compute our runtime. For the remainder of this proof we omit logarithmic factors.

Claim 4.19. *For $3 \leq i \leq 13$, $t_{k-i} = \frac{i}{2} + \frac{-8\omega^2+18\omega+18}{(3\omega+3)(i+1)} + (\omega - 2)$.*

Note that $\frac{-8\omega^2+18\omega+18}{3\omega+3} > 0$ (for any $2 \le \omega < 3$) and so Claim 4.16 holds for $\delta_{k-4}, \dots, \delta_{k-13}$.

Proof. For $i = 3$,

$$t_{k-3} = 1 + \frac{2\omega-3}{3\omega-3} + \frac{\omega^2-3\omega+3}{3\omega-3} + \frac{2\omega}{3\omega+3} = \frac{\omega^2+6\omega+3}{3\omega+3} = \frac{3}{2} + \frac{-8\omega^2+18\omega+18}{(3\omega+3)\cdot 4} + (\omega-2).$$

Therefore, by Claim 4.15, the claim holds for all $3 \le i \le 13$. ∎

Claim 4.20. *For $i \ge 13$, $t_{k-i} = \frac{i}{2} + \frac{34\omega^2-24\omega-66}{(3\omega+3)(i+1)}$.*

Again note that $\frac{34\omega^2-24\omega-66}{3\omega+3} > 0$ (for any $2 \le \omega$) and so Claim 4.16 holds for $\delta_{k-14}, \dots, \delta_0$.

Proof. For $i = 13$, by Claim 4.19,

$$t_{k-13} = \frac{13}{2} + \frac{-8\omega^2+18\omega+18}{(3\omega+3)\cdot 14} + (\omega-2) = \frac{13}{2} + \frac{34\omega^2-24\omega-66}{(3\omega+3)\cdot 14}.$$

Therefore, by Claim 4.15, the claim holds for all $i \ge 13$. ∎

For $k = 2$ these parameters are identical to the ones we chose in the proof of Theorem 4.1, taking $\delta = \delta_{k-1}, \gamma = \delta_{k-2}$. Therefore, by the calculations shown there,

$$\hat{T}_{k-2} = n^{\omega(1-\delta_{k-2},1,1-\delta_{k-2})} + n^{\omega(\delta_{k-2},1-\delta_{k-1},1)} + n^{1+\delta_{k-1}+\delta_{k-2}} \le n^{1+\omega/3}.$$

The running time for $k = 2$ is $\tilde{O}(mn^{\omega/3})$, as shown in Theorem 4.1. For $k > 2$, as computed previously,

$$\hat{T}_{k-3} = n^{\omega(2-2\delta_{k-3},1,1-\delta_{k-3})} + n^{\delta_{k-3}+t_{k-2}}.$$

Since $1 - \delta_{k-3} \le 1 \le 2 - 2\delta_{k-3}$ we have

$$\begin{aligned}\omega(2-2\delta_{k-3},1,1-\delta_{k-3}) &\le (1-\delta_{k-3}) + \delta_{k-3} + (1-\delta_{k-3})\omega = 1 + \omega(1-\delta_{k-3})\\ &= 1 + \omega\left(1 - \frac{2\omega}{3\omega+3}\right) = \frac{\omega^2+6\omega+3}{3\omega+3} = t_{k-3}.\end{aligned}$$

Thus,

$$\hat{T}_{k-3} \le n^{1+\omega(1-\delta_{k-3})} + n^{\delta_{k-3}+t_{k-2}} = n^{t_{k-3}}.$$

Now consider $k \ge 4$. By Claim 4.16, $1 - \delta_{k-i} \ge \frac{1}{2}$ and so for any $4 \le i \le 13$, $\lfloor \frac{i}{2} \rfloor \cdot (1 - \delta_{k-i}) \ge 1$. Thus,

$$\begin{aligned}&\omega\left(\left\lfloor\frac{i}{2}\right\rfloor\cdot(1-\delta_{k-i}), 1, \left\lceil\frac{i}{2}\right\rceil\cdot(1-\delta_{k-i})\right) \le \left\lfloor\frac{i}{2}\right\rfloor\cdot(1-\delta_{k-i}) - 1 + \left\lceil\frac{i}{2}\right\rceil\cdot(1-\delta_{k-i}) - 1 + \omega\\ &= i\cdot(1-\delta_{k-i}) + (\omega-2) = t_{k-i+1} + \delta_{k-i} = t_{k-i}.\end{aligned}$$

Where the final equations hold by the definition of δ_{k-i} for $4 \le i \le 13$. Therefore, for $4 \le i \le 13$, assuming the claim for $i - 1$,

$$\hat{T}_{k-i} = \mathrm{MM}(n^{\lfloor\frac{i}{2}\rfloor\cdot(1-\delta_{k-i})}, n, n^{\lceil\frac{i}{2}\rceil\cdot(1-\delta_{k-i})}) + n^{\delta_{k-i}}\cdot\hat{T}_{k-i+1} \le n^{t_{k-i}} + n^{\delta_{k-i}+t_{k-i+1}} = n^{t_{k-i}}.$$

Now consider $k \ge 14$, we again use the fact that for all $a < 0.3213$, $\mathrm{MM}(N, N^a, N) \le N^{2+o(1)}$ [VXXZ24]. For $i \ge 14$, $1 - \delta_{k-i} \ge \frac{1}{2}$ and so $\lfloor\frac{i}{2}\rfloor\cdot(1-\delta_{k-i}) \ge \frac{1}{0.3213}$. Thus for any $i \ge 14$, omitting $n^{o(1)}$ factors,

$$\begin{aligned}&\omega\left(\left\lfloor\frac{i}{2}\right\rfloor\cdot(1-\delta_{k-i}), 1, \left\lceil\frac{i}{2}\right\rceil\cdot(1-\delta_{k-i})\right) \le \left\lfloor\frac{i}{2}\right\rfloor\cdot(1-\delta_{k-i}) + \left\lceil\frac{i}{2}\right\rceil\cdot(1-\delta_{k-i})\\ &= i\cdot(1-\delta_{k-i}) = t_{k-i+1} + \delta_{k-i} = t_{k-i}.\end{aligned}$$

And so once again,

$$\hat{T}_{k-i} = \mathrm{MM}(n^{\lfloor\frac{i}{2}\rfloor\cdot(1-\delta_{k-i})}, n, n^{\lceil\frac{i}{2}\rceil\cdot(1-\delta_{k-i})}) + n^{\delta_{k-i}}\cdot\hat{T}_{k-i+1} \le n^{t_{k-i}} + n^{\delta_{k-i}+t_{k-i+1}} = n^{t_{k-i}}.$$

We conclude that for all $0 \le i \le k$, $\hat{T}_i = \tilde{O}(n^{t_i})$ and so $\hat{T}_0 = \tilde{O}(n^{t_0}) = \tilde{O}(n^{t_{k-k}})$. The final runtime of the algorithm for $k \ge 3$ is $\tilde{O}(n^{\omega}) + \hat{T}_0$. Therefore, from Claim 4.19 and 4.20 we get our desired runtimes. ∎

4.3.2 Obtaining a Runtime / Approximation Tradeoff In our proof of Theorem 4.7, we introduced an algorithm running in k steps that achieves a $2 - \frac{1}{2k-1}$ approximation to the k-radius. In each step the algorithm finds an 'approximate center', a point s_i that is close to one of the real centers c_i. However, this distance grows with each approximate center. In this section we optimize the algorithm to stop after ℓ steps, obtaining a tradeoff between approximation and runtime.

In the following theorem we demonstrate how to achieve this runtime/approximation tradeoff with a combinatorial algorithm. Similarly to Theorem 4.12, we can speed up the algorithm using fast matrix multiplication. If $\omega = 2$ we get a runtime of $\tilde{O}(n^{k-\ell+\frac{\ell(\ell+1)}{2(k+1)}})$. Using the current bounds on ω, if $\ell \leq k - 13$ we get $\tilde{O}(n^{k-\ell+\frac{\ell(\ell+1)}{2(k+1)}+o(1)})$. We omit the details for the algebraic optimization as they are similar to previous calculations.

THEOREM 4.21. *For any integer $1 \leq \ell \leq k$, there exists a combinatorial algorithm running in $\tilde{O}(mn + n^{k-\ell+\frac{\ell(\ell+1)}{2(k+1)}+1})$ time that computes a $\left(2 - \frac{1}{2\ell}, 1 - \frac{1}{2\ell}\right)$-approximation to the k-center of G with high probability.*

Proof. Let $\alpha = \left\lfloor \frac{R}{2\ell} \right\rfloor$, and γ be a parameter to be set later. Fix $c_1, ..., c_k$ to be a k-center of G. We assume that $R_k(G) \leq R$ and show how to find a set of k points that cover all vertices with radius $\leq 2R - \alpha$. As before, this allows us to distinguish between $R_k(G) > R$ and $R_k(G) \leq 2R - \alpha$, and by binary searching over possible values of R we obtain the desired approximation. We begin by computing APSP and running Algorithm 2 with different sized hitting sets, as described in Theorem 4.12. However, after ℓ steps, when we have $s_1, \ldots, s_\ell$, we stop and try every $k - \ell$ tuple in $V^{k-\ell}$ to complete the approximate k-center. See Algorithm 3 for details.

Algorithm 3 $\left(2 - \frac{1}{2\ell}\right)$-Approximation Algorithm for k-Center

```
1: Input: $G, R, k$ such that $R_k(G) \leq R$, $\ell \leq k$, parameters $\bar{\delta} = (\delta_0, \ldots, \delta_{\ell-1})$.
2: Output: Vertices $s_1, s_2, \ldots, s_k$ that cover all $V$ with radius $\leq 2R - \alpha$, where $\alpha := \left\lfloor \frac{R}{2\ell} \right\rfloor$.
3: Compute APSP.
4: STEP$(R, k, \ell, \bar{\delta}, 0, ())$.
5:
6: procedure STEP$(R, k, \ell, \bar{\delta}, i, (s_1, \ldots, s_i))$
7:     if $i < \ell$ then
8:         Sample a set $S_i$ of size $n^{1-\delta_i} \log n$.
9:         $w_{i+1} \leftarrow$ furthest vertex from $S_i$ in $B(\{s_1, s_2, \ldots, s_i\}, R + (2i-1)\alpha)^c$.
10:        $W_{i+1} \leftarrow$ closest $n^{\delta_i}$ vertices to $w_{i+1}$.
11:        if $\exists (t_{i+1}, \ldots, t_k) \in S^{k-i}$ that covers $B(\{s_1, s_2, \ldots, s_i\}, 2R - \alpha)^c$ with radius $2R - \alpha$ then
12:            return $s_1, \ldots, s_i, t_{i+1}, \ldots, t_k$.
13:        end if
14:        for $s_{i+1} \in W_{i+1}$ do
15:            STEP$(R, k, \ell, \bar{\delta}, i+1, (s_1, \ldots, s_i, s_{i+1}))$.
16:        end for
17:    else                                                    ▷ Last step, $i = \ell$
18:        if $\exists (t_{\ell+1}, \ldots, t_k) \in V^{k-\ell}$ that covers $B(\{s_1, s_2, \ldots, s_\ell\}, 2R - \alpha)^c$ with radius $2R - \alpha$ then
19:            return $s_1, \ldots, s_\ell, t_{\ell+1}, \ldots, t_k$.
20:        end if
21:    end if
22: end procedure
```

Note that Lemma 4.8 holds for any $|S| = O(n^{1-\gamma} \log n)$ and $|W| = n^\gamma$. Therefore, by Lemma 4.4 and Lemma 4.8, either we find a k-tuple that covers all V with radius $2R - \alpha$, or we will have at least one call to line 18 in which the points $s_1, \ldots, s_\ell$ satisfy $d(s_i, c_i) \leq (2\ell - 1)\alpha \leq 2R - \ell$ for every $1 \leq i \leq \ell$. Therefore, there exist $k - \ell$ points that complete this k-tuple to an approximate k-center, in particular $c_{\ell+1}, \ldots, c_k$, and so line 19 will return a k-tuple that covers the entire graph with radius $\leq 2R - \alpha$.

Runtime: We define $\bar{\delta} = (\delta_0, \ldots, \delta_{\ell-1})$ as follows. Define $t_\ell = k - \ell + 1$ and $t_i = \delta_i + t_{i+1}$ for any $i < \ell$.

- $\delta_{\ell-1} = \frac{1}{k-\ell+2}$.

- For $k-\ell+1 \le i \le k$ define $\delta_{k-i} = \frac{i-t_{k-i+1}+1}{i+1}$.

Claim 4.22. *For all $k-\ell+1 \le i \le k$, $t_{k-i} = \frac{i}{2} + \frac{(k-\ell)(k-\ell+1)}{2(i+1)} + 1$.*

Proof. For $i = k-\ell$,

$$t_\ell = k-\ell+1 = \frac{k-\ell}{2} + \frac{(k-\ell)(k-\ell+1)}{2(k-\ell+1)} + 1.$$

Therefore, by Claim 4.15 the claim holds for any $i \ge k-\ell$. ∎

Corollary 4.23. $t_0 = k-\ell + \frac{\ell(\ell+1)}{2(k+1)} + 1$.

Proof.

$$t_0 = t_{k-k} = \frac{k}{2} + \frac{(k-\ell)(k-\ell+1)}{2(k+1)} + 1 = \frac{k}{2} + \frac{(k-\ell)(k+1)}{2(k+1)} - \frac{(k-\ell)\ell}{2(k+1)} + 1$$
$$= \frac{k}{2} + \frac{k-\ell}{2} - \frac{\ell(k+1)}{2(k+1)} + \frac{\ell(\ell+1)}{2(k+1)} + 1 = k-\ell + \frac{\ell(\ell+1)}{2(k+1)} + 1.$$

∎

Denote by $\bar{T}_j$ the runtime of Step$(R, k, \ell, \bar{\delta}, j, (s_1, \ldots, s_j))$. We claim that $\bar{T}_j = \tilde{O}(n^{t_j})$.

For $j = \ell$, Step$(R, k, \ell, \bar{\delta}, (s_1, \ldots, s_\ell))$ checks for $\tilde{O}(n^{k-\ell})$ tuples if they cover a set of points. This takes $\tilde{O}(n^{k-\ell+1}) = O(n^{t_\ell})$ time.

For $j < \ell$, let $i = k-j > k-\ell$. The subroutine Step$(R, k, \bar{\delta}, j, (s_1, \ldots, s_j))$ checks for $\tilde{O}(n^{(1-\delta_j)\cdot(k-j)})$ tuples if they cover a set of points. This takes time,

$$\tilde{O}(n^{(1-\delta_j)\cdot(k-j)+1}) = \tilde{O}(n^{t_{k-i+1}+i\delta_{k-i}+1}) = \tilde{O}(n^{t_{k-i}}).$$

Further, Step$(R, k, \ell, \bar{\delta}, k-i, (s_1, \ldots, s_{k-i}))$ performs $n^{\delta_{k-i}}$ calls to Step$(R, k, \bar{\delta}, k-i+1, (s_1, \ldots, s_{k-i+1}))$ and so

$$\bar{T}_{k-i} = \tilde{O}(n^{t_{k-i}}) + \tilde{O}(n^{\delta_{k-i}+t_{k-i+1}}) = \tilde{O}(n^{t_{k-i}}).$$

The final runtime is comprised of running APSP in $O(mn)$ time and then running Step$(R, k, \ell, \bar{\delta}, 0, ())$ and so Algorithm 3 takes $O(mn) + \tilde{O}(n^{t_0}) = \tilde{O}(mn + n^{k-\ell+\frac{\ell(\ell+1)}{2(k+1)}+1})$ time. ∎

We note that the mn component of the runtime comes from computing exact APSP. To avoid this runtime, we could instead compute an additive approximation of APSP (e.g. [DHZ00] or [SY24]) and obtain a faster runtime at the expense of a greater additive error.

4.4 Approximating 3-Center In this section we refine the techniques introduced in the approximation algorithms for 2-center and general k-center for the case of 3 centers. By generalizing the ideas of Theorem 4.1 we obtain a nearly-7/4-approximation algorithm that works for graphs with integer edge weights bounded by a polynomial in n and whose running time is always less than mn for some values of m.

We first adapt Lemma 4.4 to weighted graphs with a slight change in the proof:

Lemma 4.24. *Given a graph with positive integer weights bounded by M, let S be a random sample of size $O(n^{1-\delta}\log n)$. Let w be the furthest node from S and let W be the closest n^δ nodes to w.*

For any real $r > 0$, either there exist k nodes $t_1, t_2, \ldots, t_k \in S$ that cover the graph with radius $R+r$, or w.h.p there exists some $s_1 \in W$ such that $d(s_1, c_i) < R-r+M$ for some $i \in [k]$, w.l.o.g $i = 1$.

Proof. If for every center $d(c_i, S) \le r$, denote the closest point to c_i in S by t_i. Now these k points $t_1, t_2, \ldots, t_k$ cover the entire graph with radius $R+r$, as any point $v \in V$ has $d(v, c_i) \le R$ for some $i \in [k]$, and thus $d(v, t_i) \le R+r$.

Otherwise, there exists a center such that $d(c_i, S) > r$ and therefore $d(w, S) > r$. W.h.p, the set S hits every set of size n^{δ} and so S hits W. Therefore, there exists a point $u \in W \cap S$ such that $d(w, u) > r$. By the definition of W, the set must contain all the points closer to w than u and so it contains all nodes of distance $\le r$ from w.

For some $i \in [k]$, $d(w, c_i) \le R$. Consider the shortest path between w and c_i. There are two consecutive nodes s_1 and s_1' on the path so that $d(w, s_1) \le r$ and $d(w, s_1') > r$. Because the largest edge weight in G is at most M, $d(w, s_1) > r - M$. Hence, $d(s_1, c_i) \le R - d(w, s_1) < R - r + M$. By the above observation, we have that $s_1 \in W$. ∎

Now we slightly generalize Lemma 4.10 for weighted graphs.

Lemma 4.25. *Let G be a given n-node graph with positive integer edge weights bounded by M. Let r, α be positive real numbers with $r + \alpha \le R$, let S be a random subset of vertices of size $O(n^{1-\delta} \log n)$ and $C = \{s_1, \dots, s_{k-1}\}$ be nodes such that $d(s_i, c_i) \le r$ for every $i < k$. Define $U = B(C, R + r)^c$ and $Y = B(C, 2R - \alpha)^c$. Let w be the furthest point in U from S and let W be the closest n^{δ} vertices to w. One of the following two holds.*

1. *$\exists s_k \in W$ such that $d(s_k, c_k) < r + \alpha + M$.*
2. *$Q := \bigcap_{t \in S \cap U} B(t, R) \ne \emptyset$, and any point $s_k' \in Q$ covers all points in Y with radius $2R - \alpha$.*

Proof. If $d(w, S) > R - (r + \alpha)$, then by our argument from the previous lemma, we have that $\exists s_k \in W$ such that $d(s_k, c_k) < r + \alpha + M$. Otherwise, any point $u \in U$ has $d(u, S) \le R - (r + \alpha)$.

Assume that the latter happens. Any point $t \in S \cap U$ is not covered by $c_1, \dots, c_{k-1}$ and must be covered by c_k. Thus, $c_k \in B(t, R)$ for any $t \in S \cap U$ and $c_k \in Q$, meaning $Q \ne \emptyset$.

Let s_k' be an arbitrary vertex in Q and let $y \in Y$. Let $t \in S$ be the closest vertex to y in S, by the above observation, since $Y \subseteq U$, $d(t, y) \le R - (r + \alpha)$.

If $t \notin U$, then there exists some s_i such that $d(s_i, t) \le R + r$ and hence $d(s_i, y) \le R + r + R - (r + \alpha) = 2R - \alpha$. However, this would imply $y \notin Y$ and so we conclude $t \in U$.

Therefore, $d(s_k', t) \le R$ and so $d(s_k', y) \le R + R - (r + \alpha) \le 2R - \alpha$. ∎

Let $\mu_0 = \omega - 2 + \frac{1}{\omega+1}$ and $\mu_1 = \frac{3\omega^2 - 3\omega - 1}{4\omega + 1}$. For the current bound on ω, μ_0 is about 0.6682 and μ_1 is about 0.835. For any $\omega \in [2, 3]$ we have $\mu_0 < \mu_1 < 2(\omega - 1)/3$.

Theorem 4.26. *Given an integer R and an undirected graph G with n vertices and $m = n^{1+\mu}$ edges. Suppose G has positive integer edge weights bounded by $M = poly(n)$. For $\mu \in [\mu_0, \mu_1]$, there exists an algorithm running in time*

$$\tilde{O}\left(n^{\frac{(5\omega - \omega^2 + 1) - \mu\omega(\omega - 2)}{(3\omega - \omega^2 + 1)}}\right)$$

time that w.h.p. computes a $(7/4, M)$-approximation to $R_3(G)$. For all $\mu \in [\mu_0, \mu_1]$, the running time of the algorithm is polynomially faster than both $\tilde{O}(mn)$ and $\tilde{O}(n^{\omega + 1/(\omega+1)})$.

For the current bound on ω we get that the running time is $O(n^{2.90466 - 0.35388\mu})$ for $\mu \in [0.6682, 0.835]$ and in this interval it is always better than $\tilde{O}(mn)$, the runtime of exact APSP.

Proof. We will give an algorithm that w.h.p. returns an estimate of at most $7R/4 + M$ whenever $R_3(G) \le R$. The algorithm can detect when $R_3(G) > 7R/4 + M$. The statement of the theorem follows from binary searching on R (there is a polynomial range).

Assume $R_3(G) \le R$ and let c_1, c_2, c_3 be an optimal 3-center with radius $\le R$, in this case our algorithm will find three points which cover all vertices with radius $\le 7R/4 + M$. Let S_1, S_2, S_3 be random samples of vertices in V of sizes $O(n^{1-\delta} \log n), O(n^{1-\gamma} \log n), O(n^{1-\beta} \log n)$ respectively, for parameters β, γ, δ we will set later. In

$$\tilde{O}(m(n^{1-\delta} + n^{1-\gamma} + n^{1-\beta}))$$

time we compute all the distances between $s \in S_1 \cup S_2 \cup S_3$ and $v \in V$ using Dijkstra's algorithm.

Let w_1 be the furthest node from the set S_1, let W_1 be the closest n^{δ} nodes to w_1. In time $\tilde{O}(mn^{\delta})$ run Dijkstra's from every node in W_1. By Lemma 4.24, either there exist 3 points $t_1, t_2, t_3 \in S_1$ that cover the graph with radius $R+x$ or there exists some $s_1 \in W_1$ such that $d(s_1, c_1) \le R - x + M$, for a choice of x (we will later see that $x = 3R/4$ is optimal), where s_1 covers all nodes that c_1 covers, within $2R - x + M$.

To handle the first case, construct a $|S_1| \times |V|$ binary matrix X with rows indexed by S_1 and columns indexed by V such that $X[s, v] = 0$ iff $d(s, v) \le R + x$ and a $|V| \times |S_1|^2$ binary matrix Y such that $Y[v, (s_1, s_2)] = 1$ iff both $d(v, s_1) > R + x$ and $d(v, s_2) > R + x$. Multiply X by Y in time asymptotically

$$n^{\omega(1-\delta,1,2-2\delta)} \le n^{\delta+(1-\delta)+\omega(1-\delta)} = n^{1+\omega(1-\delta)}.$$

The nodes t_1, t_2, t_3 will give $XY[t_1, (t_2, t_3)] = 0$ and any nodes that satisfy $XY[a, (b, c)] = 0$ cover all vertices with radius $R + x$.

We are now left to handle the second case, where $\exists s_1 \in W_1$ with $d(s_1, c_1) \le R - x + M$. We'll have an n^δ overhead to the runtime and assume we have $s_1 \in W_1$ such that $d(s_1, c_1) \le R - x + M$. Define $U_1(s_1) = \{u \in V : d(u, s_1) > R + x + M\}$. Note that by our choice of $x = 3R/4$, all nodes in $U_1(s_1)$ have more than $(R + x + M) - d(s_1, c_1) \ge 2x > R$ distance from c_1, and hence must be covered by either c_2 or c_3.

Let $w_2(s_1)$ be the furthest node in $U_1(s_1)$ from S_2 and let $W_2(s_1)$ be the closest n^γ nodes to $w_2(s_1)$. By the same argument as seen in the proof of Lemma 4.25, either every point in $U_1(s_1)$ is within distance $R - (2R - 2x) = 2x - R$ of S_2 or there exists a point $s_2 \in W_2(s_1)$ such that $d(s_2, c_2) \le (2R - 2x) + M$.

Case I $\forall u \in U_1(s_1)\ \exists t \in S_2$ such that $d(u, t) \le 2x - R$.

Construct a $|W_1 \times V| \times |S_2|$ binary matrix $\hat{X}$ where $\hat{X}[(s, u_1), t] = 1$ iff both $d(s, t) > 2R - x + M$ and $d(u_1, t) > R$, and a $|S_2| \times |V|$ binary matrix $\hat{Y}$ where $\hat{Y}[t, v] = 1 \iff d(t, v) > R$.

In time asymptotically

$$n^{\omega(1+\delta,1-\gamma,1)} \le n^{\delta+2\gamma+\omega(1-\gamma)} = n^{\delta+\omega-\gamma(\omega-2)}$$

compute the product $\hat{X}\hat{Y}$.

We have that $d(s_1, c_1) \le R - x + M$ so for all t such that $d(c_1, t) \le R$, $d(s_1, t) \le 2R - x + M$. Hence any point $t \in S_2$ with $d(s_1, t) > 2R - x + M$ must be covered by c_2 or c_3. We thus have $\hat{X}\hat{Y}[(s_1, c_2), c_3] = 0$.

Furthermore, we claim that any s_2, s_3 with $\hat{X}\hat{Y}[(s_1, s_2), s_3] = 0$ will cover all vertices with radius $\le R+x+M$.

Any point $u \in V$ not covered by s_1 within distance $R + x + M$ is in $U_1(s_1)$ and so by our assumption for case I there exists a point $t \in S_2$ with $d(u, t) \le (2x - R)$. Therefore, $d(s_1, t) > (R + x + M) - (2x - R) = 2R - x + M$ and since $\hat{X}\hat{Y}[(s_1, s_2), s_3] = 0$ we must have either $d(t, s_2) \le R$ or $d(t, s_3) \le R$. We conclude $d(u, s_i) \le R+x+M$ for $i = 2$ or $i = 3$.

Case II $\exists s_2 \in W_2(s_1)$ such that $d(s_2, c_2) \le (2R - 2x) + M$. This node s_2 covers all nodes that c_2 covers within radius $3R - 2x + M$.

Define $U_2(s_1, s_2) = \{u \in V : d(u, s_1) > 3R - 2x \wedge d(u, s_2) > 3R - 2x\}$.

Let $w_3(s_1, s_2)$ be the furthest in $U_2(s_1, s_2)$ from S_3 and define $W_3(s_1, s_2)$ to be the closest n^β points to $w_3(s_1, s_2)$.

We use Lemma 4.25 with parameters $\alpha = R - x$, $r = 2R - 2x$, $S = S_3, C = \{s_1, s_2\}$, where we will pick $x = 3R/4 \ge 2R/3$ so that the conditions of the lemma apply. Either $\exists s_3 \in W_3(s_1, s_2)$ such that $d(s_3, c_3) \le 3R - 3x + M$ or for any point $s_3' \in Q := \bigcap_{t \in S_3 \cap U_2(s_1,s_2)}(t, R)$ we have that s_1, s_2, s_3' cover all vertices with radius $R + x$.

To handle the first case, in total time

$$O(mn^{\beta+\gamma+\delta})$$

run Dijkstra's from every point $s_3 \in W_3(s_1, s_2)$ and check if s_1, s_2, s_3 cover all points in V with radius $R + (3R - 3x + M) = 4R - 3x + M$.

To handle the second case, construct a $|W_2(s_1)| \times |S_3|$ binary matrix $\tilde{X}$ where $\tilde{X}[s, t] = 1 \iff d(s, t) > 3R-2x$ and a $|S_3| \times |V|$ binary matrix $\tilde{Y}$ where $\tilde{Y}[t, u] = 1 \iff d(t, u) > R$.

In time asymptotically

$$n^{\delta+\omega(\gamma,1-\beta,1)} \le n^{\delta+(1-\beta-\gamma)+(1-\gamma)+\omega\gamma} = n^{\delta+2-\beta+(\omega-2)\gamma}$$

compute the product $\tilde{X}\tilde{Y}$ for every $s_1 \in W_1$. The inequality holds because we will ensure that $\gamma + \beta < 1$ and hence $\gamma < 1$ and $1 - \beta < 1$. By the same argument we have seen, for the correct s_1 we have $\tilde{X}\tilde{Y}[s_2, c_3] = 0$ and for any s_3 such that $\tilde{X}\tilde{Y}[s_2, s_3] = 0$ the triple s_1, s_2, s_3 will cover the graph with radius $R + x$.

Thus, after computing the matrix product we need to check for every $s_2 \in W_2(s_1)$ at most one potential center. This takes time $O(n^{\delta+\gamma+1})$.

Let's see what x should be set to. In the worst case the approximate radius values that we get are $4R-3x+M$, $2R-x+M$, $3R-2x+M$ and $R+x+M$.

If we set $x = 3R/4$, we get that $4R-3x = R+x = 7R/4$ and since $R-x \geq 0$ we have that $2R-x \leq 3R-2x \leq 4R-3x$. Thus the final approximate radius is at most $7R/4+M$.

Final runtime: Let $m = n^{1+\mu}$, the runtime exponents from above are:

$$\begin{aligned}
&2+\mu-\delta\\
&2+\mu-\beta\\
&2+\mu-\gamma\\
&1+\omega(1-\delta)\\
&\delta+\omega-\gamma(\omega-2)\\
&1+\mu+\beta+\gamma+\delta\\
&\delta+2-\beta+\gamma(\omega-2).
\end{aligned}$$

We will assume that

$$\gamma \leq \beta, \gamma \leq \delta, \gamma+\delta+\beta \leq 1,$$

which allows us to also conclude that $\gamma \leq \frac{1}{3} < \frac{1}{2} + \frac{\beta}{2(\omega-2)}$ and so $2-\beta+\gamma(\omega-2) < \omega-\gamma(\omega-2)$. This allows us to assume that several exponents above are dominated and what remains is:

$$\begin{aligned}
&2+\mu-\gamma\\
&1+\omega(1-\delta)\\
&\delta+\omega-\gamma(\omega-2)\\
&1+\mu+\beta+\gamma+\delta.
\end{aligned}$$

Let's set

$$\begin{aligned}
\beta &= \frac{2(\omega-1)-3\mu}{3-\omega+1/\omega}\\
\gamma &= \frac{1}{3} - \frac{\beta(\omega+1)}{3\omega}\\
\delta &= \frac{1}{3} - \frac{\beta(\omega-2)}{3\omega}.
\end{aligned}$$

We want to show that $\beta, \gamma, \delta \geq 0$, and $\gamma \leq \beta, \gamma \leq \delta, \gamma+\delta+\beta \leq 1$.

As we assumed that $\mu < 2(\omega-1)/3$ we gave that $\beta \geq 0$.

As we assumed that $\mu \geq \mu_0 = \omega - 2 + \frac{1}{\omega+1}$, we get that

$$\gamma = \frac{1}{3} - \frac{(\omega+1)}{3\omega} \cdot \frac{2(\omega-1)-3\mu}{3-\omega+1/\omega} =$$

$$\frac{1}{3} - \frac{2(\omega^2-1)-3\mu(\omega+1)}{3(3\omega-\omega^2+1)} =$$

$$\frac{(3\omega-\omega^2+1)-2(\omega^2-1)+3\mu(\omega+1)}{3(3\omega-\omega^2+1)} =$$

$$\frac{(\omega-\omega^2+1)+\mu(\omega+1)}{(3\omega-\omega^2+1)} \geq 0$$

By construction, since $\omega+1 > \omega-2$, $\delta > \gamma$.

Since we assumed that $\mu \leq \mu_1 = \frac{3\omega^2-3\omega-1}{4\omega+1}$, we also have that

$$\beta = \frac{2(\omega-1)-3\mu}{3-\omega+1/\omega} \geq$$

$$\frac{2(\omega-1)(4\omega+1)-3(3\omega^2-3\omega-1)}{(3-\omega+1/\omega)(4\omega+1)} =$$

$$\frac{8\omega^2-6\omega-2-9\omega^2+9\omega+3)}{(3-\omega+1/\omega)(4\omega+1)} =$$

$$\frac{\omega(1-\omega^2+3\omega)}{(3\omega-\omega^2+1)(4\omega+1)} = \frac{\omega}{4\omega+1}.$$

Hence:

$$\beta - \gamma = \beta - \frac{1}{3} + \frac{\beta(\omega+1)}{3\omega} =$$

$$\frac{\beta(4\omega+1)-\omega}{3\omega} \geq 0,$$

and thus $\gamma \leq \beta$, as promised.

Let's consider

$$\gamma + \delta + \beta = \frac{2}{3} + \beta \cdot \frac{3\omega-(\omega+1)-(\omega-2)}{3\omega} = \frac{2}{3} + \beta \cdot \frac{\omega+1}{3\omega}.$$

The above is ≤ 1 iff $\beta \leq \frac{\omega}{\omega+1}$. Since we have that $\mu \geq \mu_0 = \frac{\omega^2-\omega-1}{\omega+1}$, we get that

$$\beta \leq \frac{\omega(2(\omega^2-1)-3(\omega^2-\omega-1))}{(3\omega-\omega^2+1)(\omega+1)} = \frac{\omega}{\omega+1},$$

and thus $\gamma + \delta + \beta < 1$, as promised.

Now let's look at our running time exponents.

First we see that

$$1 - \beta - \delta = \frac{2}{3} + \beta\omega - 2 - 3\omega 3\omega = 2(\frac{1}{3} - \frac{\beta(\omega+1)}{3\omega} = 2\gamma,$$

and so we have that two of the exponents are equal

$$2 + \mu - \gamma = 1 + \mu + \beta + \gamma + \delta.$$

Next we see that

$$\delta(\omega+1) - \gamma(\omega-2) = \frac{\omega+1}{3} - \frac{\omega-2}{3} - \frac{\beta(\omega-2)(\omega+1)}{3\omega} + \frac{\beta(\omega-2)(\omega+1)}{3\omega} = 1,$$

And so another pair of exponents is equal:

$$1 + \omega - \omega\delta = \omega + \delta - \gamma(\omega-2).$$

Now consider the implications:

$$\frac{\omega(2(\omega-1))}{3\omega} = \frac{2(\omega-1)}{3}$$

is true and implies

$$\frac{\omega(2(\omega-1)-3\mu)+3\omega\mu}{3\omega} = \frac{2(\omega-1)}{3}$$

which by our setting of β implies

$$\mu + \frac{\beta(1-\omega^2+3\omega)}{3\omega} = \frac{2(\omega-1)}{3}$$

which implies

$$\mu + \frac{\beta(\omega+1)}{3\omega} = \frac{2\omega-2}{3} + \frac{\beta(\omega-2)}{3}$$

which implies

$$1+\mu-\frac{1}{3}+\frac{\beta(\omega+1)}{3\omega}=\omega-\frac{\omega}{3}+\frac{\beta(\omega-2)}{3}$$

and thus

$$2+\mu-\gamma=1+\omega(1-\delta).$$

Thus all four exponents are the same and they equal

$$2+\mu-\gamma=2+\mu-\frac{\omega-\omega^2+1+\mu(\omega+1)}{(3\omega-\omega^2+1)}=$$

$$\frac{(6\omega-2\omega^2+2)+(3\omega-\omega^2+1)\mu-\omega+\omega^2-1-\mu(\omega+1)}{(3\omega-\omega^2+1)}=$$

$$\frac{(5\omega-\omega^2+1)-\mu\omega(\omega-2)}{(3\omega-\omega^2+1)}.$$

□

References

[ACIM99] Donald Aingworth, Chandra Chekuri, Piotr Indyk, and Rajeev Motwani. Fast estimation of diameter and shortest paths (without matrix multiplication). *SIAM J. Comput.*, 28(4):1167–1181, 1999.

[ACLM23] Amir Abboud, Vincent Cohen-Addad, Euiwoong Lee, and Pasin Manurangsi. On the fine-grained complexity of approximating k-center in sparse graphs. In *2023 Symposium on Simplicity in Algorithms, SOSA 2023, Florence, Italy, January 23-25, 2023*, pages 145–155. SIAM, 2023.

[AGV23] Amir Abboud, Fabrizio Grandoni, and Virginia Vassilevska Williams. Subcubic equivalences between graph centrality problems, apsp, and diameter. *ACM Trans. Algorithms*, 19(1):3:1–3:30, 2023.

[AVW16] Amir Abboud, Virginia Vassilevska Williams, and Joshua R. Wang. Approximation and fixed parameter subquadratic algorithms for radius and diameter in sparse graphs. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 377–391. SIAM, 2016.

[BHW20] Maria-Florina Balcan, Nika Haghtalab, and Colin White. k-center clustering under perturbation resilience. *ACM Trans. Algorithms*, 16(2):22:1–22:39, 2020. doi:10.1145/3381424.

[BRS$^+$21] Arturs Backurs, Liam Roditty, Gilad Segal, Virginia Vassilevska Williams, and Nicole Wein. Toward tight approximation bounds for graph diameter and eccentricities. *SIAM J. Comput.*, 50(4):1155–1199, 2021.

[CFG$^+$24] Emilio Cruciani, Sebastian Forster, Gramoz Goranci, Yasamin Nazari, and Antonis Skarlatos. Dynamic algorithms for k-center on graphs. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3441–3462, 2024. doi:10.1137/1.9781611977912.123.

[CGR16] Massimo Cairo, Roberto Grossi, and Romeo Rizzi. New bounds for approximating extremal distances in undirected graphs. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 363–376. SIAM, 2016.

[CIP10] Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi. On the exact complexity of evaluating quantified k-cnf. In *Parameterized and Exact Computation - 5th International Symposium, IPEC 2010, Chennai, India, December 13-15, 2010. Proceedings*, volume 6478 of *Lecture Notes in Computer Science*, pages 50–59. Springer, 2010.

[CLR$^+$14] Shiri Chechik, Daniel H. Larkin, Liam Roditty, Grant Schoenebeck, Robert Endre Tarjan, and Virginia Vassilevska Williams. Better approximation algorithms for the graph diameter. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 1041–1052. SIAM, 2014.

[DF85] Martin E. Dyer and Alan M. Frieze. A simple heuristic for the p-centre problem. *Op. Res. Lett.*, 3(6):285—-288, 1985.

[DFHT03] Erik D Demaine, Fedor V Fomin, Mohammad Taghi Hajiaghayi, and Dimitrios M Thilikos. Fixed-parameter algorithms for the (k, r)-center in planar graphs and map graphs. In *International Colloquium on Automata, Languages, and Programming*, pages 829–844. Springer, 2003.

[DHZ00] Dorit Dor, Shay Halperin, and Uri Zwick. All-pairs almost shortest paths. *SIAM Journal on Computing*, 29(5):1740–1759, 2000. doi:10.1137/S0097539797327908.

[EKM14] David Eisenstat, Philip N Klein, and Claire Mathieu. Approximating k-center in planar graphs. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 617–627. SIAM, 2014.

[Elk05] Michael Elkin. Computing almost shortest paths. *ACM Trans. Algorithms*, 1(2):283–323, oct 2005.

[Fel19a] Andreas Emil Feldmann. Fixed-parameter approximations for k-center problems in low highway dimension graphs. *Algorithmica*, 81(3):1031–1052, 2019. URL: https://doi.org/10.1007/s00453-018-0455-0, doi:10.1007/S00453-018-0455-0.

[Fel19b] Andreas Emil Feldmann. Fixed-parameter approximations for k-center problems in low highway dimension graphs. *Algorithmica*, 81:1031–1052, 2019.

[FG88] Tomás Feder and Daniel H. Greene. Optimal algorithms for approximate clustering. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 434–444. ACM, 1988. doi:10.1145/62212.62255.

[FM20] Andreas Emil Feldmann and Dániel Marx. The parameterized hardness of the k-center problem in transportation networks. *Algorithmica*, 82:1989–2005, 2020.

[GG24] Jinxiang Gan and Mordecai J Golin. Fully dynamic k-center in low dimensions via approximate furthest neighbors. In *2024 Symposium on Simplicity in Algorithms (SOSA)*, pages 269–278. SIAM, 2024.

[GHL$^+$21] Gramoz Goranci, Monika Henzinger, Dariusz Leniowski, Christian Schulz, and Alexander Svozil. Fully dynamic k-center clustering in low dimensional metrics. In *2021 Proceedings of the Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 143–153. SIAM, 2021.

[Gon85] Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theor. Comput. Sci.*, 38:293–306, 1985. doi:10.1016/0304-3975(85)90224-5.

[HN79] Wen-Lian Hsu and George L. Nemhauser. Easy and hard bottleneck location problems. *Discret. Appl. Math.*, 1(3):209–215, 1979.

[HS86] Dorit S. Hochbaum and David B. Shmoys. A unified approach to approximation algorithms for bottleneck problems. *J. ACM*, 33(3):533–550, 1986. doi:10.1145/5925.5933.

[IP01] Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *Journal of Computer and System Sciences*, 62(2):367–375, 2001. URL: https://www.sciencedirect.com/science/article/pii/S0022000000917276, doi:10.1006/jcss.2000.1727.

[KLM19] Karthik C. S., Bundit Laekhanukit, and Pasin Manurangsi. On the parameterized complexity of approximating dominating set. *J. ACM*, 66(5):33:1–33:38, 2019. doi:10.1145/3325116.

[KLP19] Ioannis Katsikarelis, Michael Lampis, and Vangelis Th. Paschos. Structural parameters, tight bounds, and approximation for (k, r)-center. *Discret. Appl. Math.*, 264:90–117, 2019. URL: https://doi.org/10.1016/j.dam.2018.11.002, doi:10.1016/J.DAM.2018.11.002.

[Lin19] Bingkai Lin. A simple gap-producing reduction for the parameterized set cover problem. In *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPIcs*, pages 81:1–81:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. URL: https://doi.org/10.4230/LIPIcs.ICALP.2019.81, doi:10.4230/LIPICS.ICALP.2019.81.

[LVW18] Andrea Lincoln, Virginia Vassilevska Williams, and R. Ryan Williams. Tight hardness for shortest cycles and paths in sparse graphs. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 1236–1252. SIAM, 2018.

[PW10] Mihai Pătraşcu and Ryan Williams. On the possibility of faster SAT algorithms. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 1065–1075. SIAM, 2010. doi:10.1137/1.9781611973075.86.

[RV13] Liam Roditty and Virginia Vassilevska Williams. Fast approximation algorithms for the diameter and radius of sparse graphs. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 515–524. ACM, 2013.

[Sei95] R. Seidel. On the all-pairs-shortest-path problem in unweighted undirected graphs. *Journal of Computer and System Sciences*, 51(3):400–403, 1995. doi:10.1006/jcss.1995.1078.

[SY24] Barna Saha and Christopher Ye. Faster approximate all pairs shortest paths. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 4758–4827, 2024. doi:10.1137/1.9781611977912.170.

[TFL83] Barbaros C. Tansel, Richard L. Francis, and Timothy J. Lowe. Location on networks: A survey. part i: The p-center and p-median problems. *Management Science*, 29(4):482—-497, 1983.

[Tho04] Mikkel Thorup. Quick k-median, k-center, and facility location for sparse graphs. *SIAM J. Comput.*, 34(2):405–432, 2004.

[Vas18] Virginia Vassilevska Williams. On some fine-grained questions in algorithms and complexity. In *Proceedings of the ICM*, volume 3, pages 3431–3472. World Scientific, 2018.

[VXXZ24] Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. New bounds for matrix multiplication: from alpha to omega. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3792–3835, 2024. doi:10.1137/1.9781611977912.134.

[Wil05] Ryan Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theor. Comput. Sci.*, 348(2-3):357–365, 2005. URL: https://doi.org/10.1016/j.tcs.2005.09.023, doi:10.1016/J.TCS.2005.09.023.