Spectral Dual Fitting for k-Means

Aditya Anand[1], Moses Charikar[2], Vincent Cohen-Addad[3], Ruiquan Gao[2], Fabrizio Grandoni[4], Euiwoong Lee[1], Amatya Sharma[1], and Ernest van Wijland[5]

[1]University of Michigan, {adanand,euiwoong,amatya}@umich.edu
[2]Stanford University, {moses,ruiquan}@cs.stanford.edu
[3]Google Research, cohenaddad@google.com
[4]IDSIA, USI-SUPSI, fabrizio.grandoni@gmail.com
[5]Université Paris-Cité, ernest.vanwijland@irif.fr

Abstract

We give a new dual fitting algorithm which gives improved approximation ratios of $3 + \ln 2 + \varepsilon$ (≈ 3.694) and $4.9 + \varepsilon$ for k-Means in (high-dimensional) Euclidean and general metrics respectively, improving upon the previously known ratios of $4 + \varepsilon$ [Charikar, Cohen-Addad, Gao, Grandoni, Lee, and van Wijland STOC'26] and $5 + \varepsilon$ [Byrka, Guo, Hu, Li, Wan, Wang FOCS'26], resp. In particular, our result for Euclidean k-Means breaks the hardness barrier of $1 + 8/e \approx 3.94$ for Metric k-Means. Prior to our work, no such separation between general and Euclidean metrics was known for k-Median, k-Means, or Facility Location in terms of their approximability.

Unlike prior dual fitting approaches for k-Means, our new dual fitting algorithm tightly accounts for dual payments while still facilitating an effective dual feasibility analysis. We introduce a new framework that uses spectral analysis for determining the approximation factor of our algorithm.

Contents

1 Introduction **1**

2 Technical Overview **3**
2.1 Spectral analysis: the general framework proving the dual feasibility 8
2.2 Spectral analysis for the Euclidean case 10
2.3 Spectral analysis for the metric case 12
2.4 Turning LMP approximations to bicriteria approximations 13

3 Preliminaries **13**

4 Our Improved Greedy Algorithm **15**

5 Establishing the LMP Approximations **21**
5.1 Proof Roadmap 22
5.2 Defining the parameters 23
5.3 Proof of Lemma 4.11 25
5.3.1 Step 1: Bounding the extra cost term via $P_{j\ell}$ and d_j^+ 25
5.3.2 Step 2: Bounding the Laplacian matrix of P 26
5.3.3 Step 3: A key property on the conditional column sum of P 27
5.3.4 Step 4(a): Proof for the Euclidean case 30
5.3.5 Step 4(b): Proof for the general metric case 32
5.4 A generalization of Lemma 4.11 for the Euclidean case 33

6 An Improved LMP 4.9-Approximation for the Metric Case **34**
6.1 Unweighted 34
6.2 Weighted 41

7 Log-Adaptive Algorithm for Weighted Facility Location **44**
7.1 Dual Payment 46
7.2 Dual Feasibility 47

8 Approximation with $O(\log n/\varepsilon^3)$ Extra Centers **59**
8.1 Pre-processing 59
8.2 Walking Between Two Solutions: Setup 60
8.3 MergeSolutions 66
8.3.1 Making Parameters Identical 68
8.3.2 Maximize Common Prefix of H_p and H'_p 71
8.3.3 Removing Extra Facilities 74
8.3.4 Making the Probability of Direct Connects Almost Identical 76
8.4 Analysis of the Robust Algorithm 79
8.4.1 Dual Payment 84
8.4.2 Dual Feasibility 86

9 $(4+O(\sqrt{\varepsilon}))$-Approximation for $(\frac{\zeta}{\log n})$-Stable Instances **98**
9.1 Properties of a Locally Optimal Solution 99
9.2 Pure Clusters 100
9.3 Cheap Clusters and D-Sample Process to Hit Expensive Clusters 100

9.4 Identifying Balls of Expensive Clusters . . . 101
9.5 *Dummy* Centers and Mixed Solutions M_O and M_D . . . 102
9.6 Removal of Expensive Centers of S_0 . . . 103
9.7 Consistently Assigning Clients in Mixed Solutions and the sets $\mathcal{U}$ and $\mathcal{R}$ . . . 103
9.8 Removal of Cheap Centers in S_0 . . . 109
9.9 Mixed Solutions after the Removal of $\widetilde{\mathcal{U}}$ . . . 109
9.10 Finding S^* via Submodular Optimization . . . 112
9.11 Putting Everything Together – Proof of Theorem 1.4 . . . 115

10 $(2+O(\sqrt{\varepsilon}))$-Approximation for $(\frac{\zeta}{\log n})$-Stable Euclidean Instances 117
10.1 Locally Optimal Solution and Pure Clusters . . . 118
10.2 Leaders' Sampling of Leader-Expensive Clusters of OPT . . . 119
10.3 Removal of Expensive Centers of S_{disc} . . . 121
10.4 Removal of Cheap Centers from S . . . 122
10.5 Putting Everything Together – Proof of Theorem 1.5 . . . 123

A Proof of Theorems 1.1 and 1.2 127

B Missing Proofs in Section 5 128
B.1 Proof of Lemma 5.8 . . . 128
B.1.1 Preliminaries on weighted k-means . . . 128
B.1.2 The proof . . . 130
B.2 Proof of Lemma 5.20 . . . 132

C Missing Proofs in Section 6 135
C.1 Proof of Lemma 6.6 . . . 135
C.2 Weighted: Full Proof . . . 138

1 Introduction

Clustering is one of the most fundamental tasks in data science with wide-ranging applications across various domains, including unsupervised learning, text and image analysis, recommender systems, and bioinformatics. (Metric) k-Means is one of the most well-studied clustering objectives. Given a collection of n clients D, a collection F of potential center locations in a metric space defined by distances $d(.,.)$, and an integer $k \in [n]$, the goal is to select k centers $S \subseteq F$ such that $\text{conn}(S) := \sum_{j \in D} d^2(j, S)$ is minimized, where $d(j, S)$ is the distance of point j to its nearest center in S. A very important special case is *Euclidean k-Means*, where D are points in a Euclidean space, and F is implicitly given by any point in the Euclidean space.[1] By standard techniques [Mat00; DKKR03] and losing only a $(1 + \varepsilon)$ factor in the approximation ratio for any constant $\varepsilon > 0$, we can restrict F to a polynomial-time-computable subset of points. W.l.o.g., we will also assume that $D \subseteq F$.

From a theoretical standpoint, the study of clustering problems through the lens of (polynomial-time) approximation algorithms has been extremely successful, yielding new algorithms and analysis techniques. In particular, k-Means and the closely related k-Median (whose objective function $\sum_{j \in D} d(j, S)$ is the sum of non-squared distances) are two problems that have received the most attention. For k-Median, there have been gradual developments over the last three decades using various algorithmic techniques, including primal LP rounding [CGTS99; CL12], primal-dual [JV01], greedy algorithm with dual fitting analysis [JMS02; JMM+03; CGLS23], local search [AGK+04], and bipoint rounding [LS13; BPR+15; GPST23]. The current best approximation ratio is $(2 + \varepsilon)$, achieved by [CGL+25]. It is known that one cannot improve the approximation ratio beyond $1 + 2/e \approx 1.73$ [GK99].

For k-Means, the picture has been much more elusive. While new algorithmic techniques for k-Median often translate to k-Means, it takes nontrivial technical effort, and the approximation ratios usually end up being much larger: this is because only approximate versions of triangle inequalities are available for squared metrics, which present technical challenges for some k-Median algorithms, most notably the ones based on dual fitting.

For Metric k-Means, [GT08] gave a 25-approximation using local search, which was improved to a $(9 + \varepsilon)$ approximation by [ANSW20] via primal-dual. Recently, the approximation factor was improved to 5.83 [CCG+25] via a novel dual fitting algorithm. More recently, Byrka et al. [BGH+26] introduced a new LP rounding approach that improves the ratio to $5 + \varepsilon$. For Euclidean k-Means, [KMN+04] gave a $(9 + \varepsilon)$-approximation using local search, followed by improvements using primal-dual methods [ANSW20; GOR+22; CEMN22] and most recently to $4 + \varepsilon$ [CCG+26] via dual fitting. The hardness for the metric case is $1 + 8/e \approx 3.94$, while for Euclidean metrics the hardness is 1.17 [CK19], and under a stronger conjecture called the Johnson Coverage Hypothesis (JCH), the hardness can be strengthened to 1.73 [CSL22]. We remark that we focus on the high-dimensional Euclidean case here, while better algorithms are known for the low-dimensional one [FRS16; CKM16; Coh18].

F: Is this 1+2/e? If yes pls add

In this work, we give a new dual fitting based algorithm that gives significantly better approximation ratios for both Euclidean k-Means and Metric k-Means. Our algorithm leads to a $3 + \ln 2 + \varepsilon$ (≈ 3.694) approximation for the Euclidean case and a $4.9 + \varepsilon$ approximation for the metric case. Our result for Euclidean metrics is the first algorithm that gives an approximation ratio *below the* 3.94*-hardness barrier* for the metric case, showing that the optimal approximation ratios for these two versions will be different assuming $\mathbf{P} \neq \mathbf{NP}$. Such a separation between general metrics and high-dimensional Euclidean metrics, even after

[1] Sometimes this is called the *continuous* Euclidean k-Means problem in the literature.

extensive recent work [CEMN22; CSL22; LS25], has not been achieved for k-Median and Facility Location.

Theorem 1.1. *For any $\varepsilon > 0$, there is a polynomial-time algorithm that yields a $(3+\ln 2+\varepsilon)$-approximation for Euclidean k-Means.*

Theorem 1.2. *For any $\varepsilon > 0$, there is a polynomial-time algorithm that yields a $(4.9+\varepsilon)$-approximation for Metric k-Means.*

Our main technical contribution is an improved Lagrangian Multiplier Preserving (LMP) approximation for the following (Uncapacitated) Facility Location problem (with squared distances and uniform opening costs), which we simply call *Facility Location* throughout the paper: the input consists of a set of clients D, a collection of centers F, a (possibly Euclidean) metric d on $D \cup F$, and a *facility opening cost* f, and the goal is to select $S \subseteq F$ that minimizes $f|S| + \mathrm{conn}(S)$. In particular, $S \subseteq F$ is called an LMP c-approximation if $c \cdot f|S| + \mathrm{conn}(S) \leq c \cdot \mathrm{OPT}_{\mathrm{LP}}(f)$, where $\mathrm{OPT}_{\mathrm{LP}}(f)$ denotes the optimal cost of the standard LP relaxation for Facility Location defined in Section 3. Our algorithms yield LMP 4.9 and $(3+\ln 2)$-approximations for Facility Location in general and Euclidean metrics, respectively.

Our improved LMP approximation is obtained via a refined dual fitting algorithm that tightly accounts for the cost incurred, while still being able to facilitate an improved one-cluster based dual feasibility analysis. We also introduce a novel spectral analysis for k-Means, departing from the usual factor-revealing linear programs. The spectral analysis provides a more accurate method to analyze the squared objective in k-Means compared to the previous approaches. More specifically, upper bounding the approximation factor reduces to upper bounding the largest eigenvalue of a family of matrices. The latter task is highly non-trivial, but it leads to substantially lower approximation factors compared to known methods.

This is then combined with the two-step framework recently used in [CGL+25; CCG+25; CCG+26] to obtain our approximation algorithm. For the first step, building on our LMP approximation, we use the idea of *log-adaptive algorithms* to obtain a good approximate solution for k-Means that opens $k + O(\log n/\varepsilon^3)$ centers. Let OPT_k be the optimal cost for an instance of k-Means.

Theorem 1.3. *For any constant $\varepsilon > 0$, there is a polynomial-time algorithm that, given an instance of k-Means, finds $S \subseteq F$ with $|S| = k + O(\log n/\varepsilon^3)$ satisfying $\mathrm{conn}(S) \leq (\Gamma + O(\sqrt{\varepsilon})) \cdot OPT_k$, where $\Gamma = 3+\ln 2$ for Euclidean metrics and $\Gamma = 4.9$ for general metrics.*

The second step of the framework studies *stable instances*. A k-Means instance is β-stable if $\mathrm{OPT}_{k-1} \geq (1+\beta)\mathrm{OPT}_k$, where OPT_{k-1} denotes the optimal cost of $(k-1)$-Means with the same clients and centers. We show that a slightly more careful analysis of the same algorithms as in [CCG+25; CCG+26] yields the following results.

Theorem 1.4. *For any constants $\varepsilon, \zeta > 0$, there is a polynomial-time randomized algorithm for Metric k-Means that with high probability returns a solution of cost at most $(4+\varepsilon)OPT_k$ assuming that the input instance is $(\zeta/\log n)$-stable.*

Theorem 1.5. *For any constants $\varepsilon, \zeta > 0$, there is a polynomial-time randomized algorithm for Euclidean k-Means that with high probability returns a solution of cost at most $(2+\varepsilon)OPT_k$ assuming that the input instance is $(\zeta/\log n)$-stable.*

These results improve the $(5+\varepsilon)$-approximation for general metrics and the $(4+\varepsilon)$-approximation for Euclidean metrics [CCG+25; CCG+26]. Similar to previous results [CGL+25; CCG+25; CCG+26], given Theorems 1.3 to 1.5, we can easily prove our main results Theorem 1.1 and Theorem 1.2; given a k-Means instance, if it is not $\Omega(\varepsilon^4/\log n)$-stable as a k'-Means instance for each $k' = k, k-1, \ldots, k-O(\log n/\varepsilon^3)$, then $\mathrm{OPT}_{k-O(\log n/\varepsilon^3)} \approx \mathrm{OPT}_k$ and we can use Theorem 1.3 with $k \leftarrow k - O(\log n/\varepsilon^3)$. Otherwise, the instance is stable for some k' in the above range, and we can use Theorems 1.4 and 1.5. The formal proof appears in Section A.

Concurrent work. Concurrently and independently, Byrka, Guo, Hu, Li, Wan, and Wang [BGH+26] gave an iterative randomized rounding framework for k-clustering with ℓ_p^p-type assignment costs. For Metric k-Means, their framework gives a $(5+\varepsilon)$-approximation, improving the previous 5.83-approximation. For Euclidean k-Means, they give a $(4+\varepsilon)$-approximation and prove an 11/3-LMP approximation. Their approach is LP-rounding based and is conceptually different from our dual-fitting/spectral analysis. Our results have better approximation guarantees for both settings, giving $(4.9+\varepsilon)$ for Metric k-Means and $(3+\ln 2+\varepsilon)$ for Euclidean k-Means.

F: In view of this paragraph, I wonder whether we should remove the earlier citations of Byrka et al. (including footnote 2)

2 Technical Overview

We start by revisiting the dual fitting algorithm for Facility Location of the two papers by Charikar et al. [CCG+25; CCG+26] for Metric and Euclidean k-Means, respectively. The two algorithms are very similar, except for one difference that [CCG+26] lowers certain dual values which can lead to further improvements. In this overview, this difference is not important, but we will use the stronger algorithm from [CCG+26] which is still as good as the older algorithm [CCG+25] for the metric case as well.

We provide a description of their algorithm below.

Greedy Algorithm. The algorithm maintains a set A of active clients, a set $S \subseteq F$ of open facilities, and a partition of clients into three classes: active clients A, indirectly connected clients IC, and directly connected clients DC. Intuitively, active clients are not yet connected to an open facility, and they are actively growing their dual variables. Indirectly connected clients are connected to an open facility, but they do not directly contribute (a strictly positive amount) to the opening cost of that facility. On the other hand, directly connected clients are connected to an open facility and they directly contribute to the opening cost of that facility. Initially, $A = D$, $IC = DC = S = \emptyset$, and $\alpha_j = 0$ for every client $j \in D$. Fix a parameter $\rho > 1$. We let $\hat{f} = \Gamma f$ where Γ is some constant that depends on ρ to be specified later, and f is the (uniform) facility cost.

As long as $A \neq \emptyset$, the algorithm increases uniformly the (*dual*) values α_j for all clients $j \in A$ until one of the following two events occurs.

1. A facility becomes tight. There exists an unopened facility $i \in F \setminus S$ such that

$$\sum_{j \in A \cup IC} \left[\alpha_j - \rho \cdot d^2(j,i)\right]^+ + \sum_{j \in DC} \left[\rho d^2(j,S) - \rho d^2(j,i)\right]^+ = \hat{f}. \tag{1}$$

   In this case, the algorithm opens facility i. Every client $j \in DC$ with $d^2(j,S) > d^2(j,i)$ is reconnected to i. For every client $j \in A \cup IC$, the algorithm acts as follows:

- if $\alpha_j \geq \rho \cdot d^2(j,i)$, then j is connected (or reconnected) to i and moved to DC;
- else, if $\alpha_j \geq d^2(j,i)$, then j is connected (or reconnected) to i, its value α_j is lowered to $d^2(j,i)$, and j is moved to IC if it was in A.

2. An active client reaches its current service cost. For some client $j \in A$, we have

$$\alpha_j = d^2(j,S).$$

In this case, the client j is moved to IC and connected to its closest facility in S.

Intuitively, the quantities of type $[\cdot]^+$ above are *bids* that client j is willing to pay to open facility i. Via these bids, j can contribute to the opening of multiple facilities. At the same time, the residual dual α_j of j after subtracting these bids is sufficient to connect j to the closest open facility. The algorithm naturally defines a notion of *time*, such that at time θ, all the active clients have α-value θ.

Then, they show the following two cost accounting lemmas (similar to [JMM+03]) for this algorithm (considering the values of α and S at the end of the algorithm), that lead to an LMP Γ-approximation.

Lemma 2.1 (Dual Payment, for any metric)**.** $\sum_{j \in D} \alpha_j \geq \hat{f}|S| + \sum_{j \in D} d^2(j,S)$.

Note that the RHS of this inequality is the cost of the computed solution (with facility cost $\hat{f} = \Gamma f$). The proof of this lemma follows from the fact that a facility is only opened when it is fully paid for (by the bids), and the fact that for any client j, after subtracting any bid to open facilities, the remaining part of α_j can still pay for its connection cost $d^2(j,S)$.

Lemma 2.2 (Dual Feasibility, [CCG+25; CCG+26])**.** *Let $\rho = \sqrt{2}+1$ (resp., $\rho = 2$ for Euclidean metrics). Fix any facility i. Then*

$$\sum_{j \in D} \left[\alpha_j - \Gamma d^2(j,i)\right]^+ \leq \hat{f},$$

where $\Gamma = 3 + 2\sqrt{2}$ (resp., $\Gamma = 4$ for Euclidean metrics).

Lemma 2.2 implies that (α/Γ) is dual-feasible for the Facility Location LP (see Dual FL LP) with opening cost f. Combining this with Lemma 2.1, one gets

$$\Gamma f|S| + \mathrm{conn}(S) \leq \sum_{j \in D} \alpha_j \leq \Gamma \cdot \mathrm{OPT}_{LP}(f),$$

concluding that S is an LMP Γ-approximate solution.

While the dual payment lemma more or less follows directly from the algorithm description, the key challenge in [CCG+25; CCG+26] has been proving dual feasibility given the complexity of the algorithm due to the presence of both IC and DC clients and the non-monotonicity of the dual variables.

The barrier of previous works. Our first observation is that there is a barrier for the dual feasibility analysis of the algorithm of [CCG+26] for Euclidean metrics — a very simple example shows that the LMP 4-approximation analysis of Lemma 2.2 can be tight for some facility i. As illustrated in Figure 1, we will consider points on a line (i.e., 1-dimensional Euclidean space). Let i be a facility and $D^* = \{1,2\}$ located at $x_i = 0$, $x_1 = -1$, $x_2 = 1$.

Then, $d^2(1,i) + d^2(2,i) = 2$. Let $\hat{f} = 20$. The algorithm simultaneously increases α_1 and α_2 until $[\alpha_1 - \rho d^2(1,i)]^+ + [\alpha_2 - \rho d^2(2,i)]^+ = [\alpha_1 - 2]^+ + [\alpha_2 - 2]^+ = 20$, i.e., when $\alpha_1 = \alpha_2 = 12$. At this time point, suppose that client 1 gets directly connected to the only open facility at $x_{i'} = -3$ (which was opened via bids from client 1 and other clients not in D^*). This leads to $d^2(1,S) = 4$. From now on, the bid of client 1 to facility i becomes $8 - 2 = 6$. We will keep increasing α_2 until one of the following holds:

1. $\alpha_2 \geq d^2(2,S) = (x_2 - x_{i'})^2 = 16$. In this case, we have client 2 indirectly connected to i'.
2. $6 + [\alpha_2 - 2]^+ = 20$, which means $\alpha_2 = 16$. In this case, we have i opened and client 2 directly connected to i.

Therefore, we have $\sum_{j \in D^*} \alpha_j = 12 + 16 = 28$. On the other hand, we have $\hat{f} + 4(d^2(1,i) + d^2(2,i)) = 20 + 4 \cdot 2 = 28$. Hence, the dual feasibility is tight for this example, and we cannot hope for an LMP approximation ratio below 4 using this algorithm.

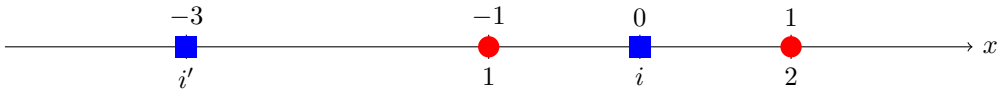


Figure 1: Barrier example for the previous dual-feasibility analysis. The facility i is at 0, the two clients are at -1 and 1, and an already open facility i' is at -3. Thus $d^2(1,i) = d^2(2,i) = 1$, while $d^2(1,i') = 4$ and $d^2(2,i') = 16$.

An issue with the payments. Our new algorithm is inspired by the fact that while the dual feasibility can be tight, the dual payment Lemma 2.1 does not tightly account for the costs: $\sum_j \alpha_j$ may actually be larger than the cost of the algorithm. In particular, we observe that for any client that is directly connected at the end, the algorithm always overpays the connection cost by a factor of ρ. To see this, let DC^{end} denote the set of clients which are directly connected at the end. Let $j \in DC^{end}$, and suppose through the execution of the algorithm, j was connected to facilities $i_1, i_2 \ldots \ldots i_t$. For simplicity, assume that j was directly connected to each of these facilities. Then the total bid (dual contribution, see Equation (1)) of j to these facilities is at most

$$(\alpha_j - \rho d^2(j,i_1)) + \sum_{\ell=1}^{t-1} \rho d^2(j,i_\ell) - \rho d^2(j,i_{\ell+1}) = \alpha_j - \rho d^2(j,i_t) \ .$$

This means that after accounting for facility costs, α_j pays for $\rho d^2(j,i_t)$, where $\rho > 1$, leading to an overpayment of the connection cost by the factor ρ. Unfortunately, one cannot set $\rho = 1$ as it would not give even an LMP $O(1)$-approximation, as already observed in [CCG+25]. This means we could simply subtract $(\rho - 1)d^2(j,i_t)$ in the dual feasibility analysis.

However, while this overpayment is not ideal, it seems challenging to account for this naturally in the analysis: dual feasibility analysis is highly *local*, which cannot rely on the sequence of connections a client has: in particular, it is not clear how to meaningfully incorporate the subtraction $(\rho - 1)d^2(j,i_t)$ of client j, which is a *global* quantity, into the local one-cluster dual feasibility analysis. The previous example of Figure 1 illustrates this difficulty as well; when $\alpha_2 = 16$, the events triggering the actions of "having client 2 indirectly connected to i'" and "opening i and connecting client 1 and 2 to i" happen simultaneously, but depending on how the algorithm breaks the ties (which can be easily manipulated by

the position of the external facility i'), the amount of the overpayment substantially changes. In the former case, it is $(\rho-1)\cdot d^2(1,i')=4$ just from client 1, and in the latter case, it is $(\rho-1)\cdot(d^2(1,i)+d^2(2,i))=2$.

Our new algorithm with tight dual payments. To resolve this issue, we propose a new algorithm (Algorithm 1) that allows us to get the best of both worlds. Our new algorithm still has an overpayment, but we subtract this overpayment at the end, while still being able to incorporate this subtraction in the dual feasibility analysis. In our new algorithm, each client *reserves* a part of its first direct connection cost, that it does not offer to other facilities for subsequent (direct) connections. The key is that clients only move to facilities where their connection cost is lower, so the first direct connection cost is an *upper bound* for *every* future direct connection.

Algorithm 1 (Greedy Algorithm).

Initialization: $A=D$, $IC=DC=S=\emptyset$, and $R_j=\alpha_j=0$ for every $j\in D$.

While $A\neq\emptyset$, uniformly grow the α_j of every client in A until:

1. For some unopened facility $i\in F\setminus S$, we have

$$\sum_{j\in A\cup IC}\left[\alpha_j-\rho\cdot d^2(j,i)\right]^+ + \sum_{j\in DC}\left[d^2(j,S)-d^2(j,i)\right]^+ = \hat{f}.$$

   In this case, we open facility i. For each client $j\in DC$, if $d^2(j,S)>d^2(j,i)$, we reconnect j to i. For each client $j\in A\cup IC$,

   - If $\alpha_j\geq\rho\cdot d^2(j,i)$, we (re)connect j to i, move j to DC and set $R_j=d(j,i)$.
   - Else, if $\alpha_j\geq d^2(j,i)$, we (re)connect j to i, and move j to IC (if $j\in A$), and set $\alpha_j=d^2(j,i)$.

2. For some client $j\in A$, we have

$$\alpha_j=d^2(j,S).$$

   In this case, we move j to IC and connect j to the closest facility to it in S.

Notice that until the first direct connection, an active or IC client j offers $\alpha_j-\rho d^2(j,i)$, but subsequently, instead of offering the natural $\rho d^2(j,S)-\rho d^2(j,i)$, it only offers $d^2(j,S)-d^2(j,i)$. Thus if $R_j=d(j,i_1)$ is the first direct connection distance, then $(\rho-1)R_j^2$ is *reserved* by the client and never used for any subsequent bid to facilities.

Following the same analysis as before, consider any client $j\in DC^{end}$. If it goes through a sequence of direct connections $i_1,i_2,\ldots,i_t$, then it offers

$$\alpha_j-\rho d^2(j,i_1)+\sum_{\ell=1}^{t-1} d^2(j,i_\ell)-d^2(j,i_{\ell+1}) = \alpha_j-(\rho-1)d^2(j,i_1)-d^2(j,i_t)\ .$$

Since the connection cost at the end is only $d^2(j,i_t)$, this leads to an overpayment of $(\rho-1)d^2(j,i_1)$, where i_1 is the first facility j was directly connected to. But now we define

$R_j = d(j, i_1)$, and subtract $(\rho - 1)R_j^2$ in the dual feasibility analysis to make this accounting exact (we will define $R_j = 0$ if j is never directly connected). This turns out to be helpful in the dual feasibility analysis because we know $R_j = d(j, i_1) \geq d(j, i_2) \geq \dots \dots \geq d(j, i_t)$, as clients only move to closer facilities as the algorithm progresses. R_j is thus an upper bound on all these distances, which facilitates a refined dual feasibility analysis, as discussed in greater detail in the following subsections.

Going back to the example from Figure 1 with $\rho = 2$, we set $R_1 = d(1, i') = 2$ when i' is open at time 12, and the new algorithm indirectly connects client 2 to i' at time 16 (which now happens regardless of the tiebreaking rule as the bid of client 1 toward i becomes smaller), leading to $R_2 = 0$. Thus we have $\sum_{j \in D^*} \alpha_j = \alpha_1 + \alpha_2 = 28 = \hat{f} + 4\big(d^2(1, i) + d^2(2, i)\big)$ just like before, but because of the subtraction of $(\rho - 1)R_1^2 = R_1^2$, the dual feasibility is no longer tight. In particular we have $\alpha_1 - R_1^2 + \alpha_2 - \hat{f} = 4 = 2\big(d^2(1, i) + d^2(2, i)\big)$, and consequently, $\sum_{j \in D^*}(\alpha_j - 2d^2(j, i) - R_j^2)^+ \leq \hat{f}$, which improves the dual feasibility factor to 2 instead of 4 for this example.

Analysis. From the above discussion, our modified payment and dual feasibility lemmas can be stated as follows. Let α and S be the respective values at the end of the algorithm. The following dual payment lemma almost directly follows from the algorithm description and the discussion above.

Lemma 2.3 (Dual Payment for our new algorithm, Lemma 4.5)**.**

$$\sum_{j \in D} (\alpha_j - (\rho - 1)R_j^2) \geq \hat{f}|S| + \sum_{j \in D} d^2(j, S)$$

The crux of our analysis is now to prove a dual feasibility lemma for a small enough constant Γ (depending on ρ). In particular, we prove the following.

Lemma 2.4 (Dual feasibility of our new algorithm, Lemma 4.6)**.** *Fix any facility i. Let $\rho = 2.5$ (resp., $\rho = 2$ for Euclidean metrics). Then*

$$\sum_{j \in D} \left[\alpha_j - \Gamma d^2(j, i) - (\rho - 1)R_j^2\right]^+ \leq \hat{f},$$

where $\Gamma = 4.9$ (resp., $\Gamma = 3 + \ln(2)$ for Euclidean metrics).

F: The claim looked odd without specifying ρ and Γ. Formatted like Lemma 2.2. Now it is essentially equal to Lemma 4.6, so we might rather use restatable. Notice that 2.2 and 4.6 wrongly have the same label

Intuitively, the above dual feasibility lemma shows that the variables $\alpha'_j := \frac{1}{\Gamma}(\alpha_j - (\rho - 1)R_j^2)$ induce a dual-feasible solution for the Facility Location LP. An LMP Γ-approximation follows by the same argument as before.

Furthermore, note that we can interpret our algorithmic idea as having smaller bids for clients who are directly connected. The intuition behind our new algorithmic idea (for the Euclidean case) is that these directly connected clients are "bad" in the dual feasibility analysis of [CCG+26] only when all clients and facilities are "almost" on a line (i.e., one-dimensional Euclidean space) and its maximum direct connection cost R_j^2 is close to the squared distances between it and other clients j' with large dual variable $\alpha_{j'}$ (compared to $d^2(j', i)$). As we have this additional $-(\rho - 1)R_j^2$ term in the final dual variable α'_j, we can make the final dual variables of these "bad" directly connected clients much smaller.

AS: Lemma 4.6 is never referred to anywhere. so for now just changing its label. the statements of Lemma 4.6 and this Lemma are slightly different, because there's no α^* here.

However, to apply this intuition to the analysis (for the general Euclidean cases), we confront several technical challenges:

- We need to deal with high-dimensional Euclidean spaces.
- We need to transform these $-(\rho-1)R_j^2$ terms into $-d^2(j,i)$ terms (because, intuitively, the dual feasibility constraint compares $\sum_{j\in D'}\alpha_j-(\rho-1)R_j^2$ with $f+\sum_{j\in D'}\Gamma d^2(j,i)$ for every $D'\subseteq D$).
- Finally, in the Euclidean space, [CCG+26] uses the pairwise distance $\frac{1}{|D'|}\sum_{j<\ell\in D'}d^2(j,\ell)$ as an intermediate quantity, which is upper bounded by $\sum_{j\in D'}d^2(j,i)$ (see, e.g., Lemma B.3). However, this inequality may not hold if the coefficients before each $d^2(j,i)$ have an average of 1 but are "unbalanced". After we transform the $-(\rho-1)R_j^2$ terms into the $-d^2(j,i)$ terms, we observe that the coefficients of the new terms have an average $\Omega(1)$ in the cases [CCG+26]'s analysis is tight. However, the coefficients are "unbalanced", and some of them can even be zero (due to $R_j=0$ for the corresponding client). To prove a ratio strictly better than 4, the most significant challenge is to show that the pairwise distance terms combined with these new $-d^2(j,i)$ terms can be upper bounded by $(1-\Omega(1))\cdot\sum_{j\in D'}d^2(j,i)$.

As an illustration on how we resolve these challenges, we will provide a framework of our spectral analysis in Subsection 2.1, and provide simple proofs that obtain improved ratios (which are slightly worse than our final results) on the worst cases for the analysis of [CCG+26; CCG+25] in Subsections 2.2 and 2.3.

We remark that slightly smaller values of Γ (hence slightly better approximation factors)[2] can be obtained, for both the Euclidean and the metric case, with different values of ρ and a substantially more technical and longer analysis. We decided to omit this for the sake of clarity (and given that the presented analysis is already very complex). We also remark that our approach cannot yield significantly better approximation factors. Our numerical simulations suggests that our spectral analysis framework has a barrier of 3.51 for Euclidean metrics and 4.67 for general metrics. See the end of Subsection 2.1 for a brief discussion of this.

2.1 Spectral analysis: the general framework proving the dual feasibility

In this part, we will introduce the general idea of our new spectral analysis of the dual feasibility, which is the key to improving the approximations for both Euclidean and Metric k-Means. For simplicity, we will focus on the special case where at each point in time each considered client is active or directly connected (without ever being indirectly connected). This case is already non-trivial, and allows us to illustrate the main ideas. To simplify the notation, we also assume that the final dual values of the mentioned clients are all distinct. We start with some general considerations. Then, in Subsection 2.2 and Subsection 2.3 we will detail the analysis (for the mentioned special case) in the Euclidean and metric case, respectively. The analysis for arbitrary instances is given in Sections 4 and 5 for the Euclidean case, and in Sections 4 to 6 for the metric case.

The general idea of the spectral analysis. We will consider a similar set of constraints on the dual variables α_j as in the original analysis of [JMM+03] as the starting point of our spectral analysis. Similar constraints are also used in the analysis of [CCG+25; CCG+26].

[2] For example, we know that $\rho=1.96,\Gamma=3.692$ works for Euclidean metrics, and $\rho\approx 2.430,\Gamma=4.865$ works for general metrics.

Fix a facility i, and let $D^* = \{j \mid \alpha_j - (\rho-1)R_j^2 \geq \Gamma d^2(j,i)\}$. We need to show that

$$\sum_{j \in D^*} \alpha_j - (\rho-1)R_j^2 \leq \hat{f} + \Gamma \sum_{j \in D^*} d^2(j,i) \ . \tag{2}$$

Let us assume by renumbering that $j \in D^*$ is sorted in increasing order of α_j, and let us denote $D^* = [s]$. Recall that we assume that no two α_j are equal.

EL: Remind that, since we are assuming everything DC, α_j and the first connection time is the same. This is always true, right?

Then we consider the contributions to i at time $\alpha_t - \varepsilon$ for some arbitrarily small $\varepsilon > 0$ for each $t \in [s]$. Observe that each client $j < t$ is already directly connected to an open facility in our special case. Let S^t be the set of open facilities at time $\alpha_t - \varepsilon$, and $r_{jt} = d(j, S^t)$ be the distance at which j is connected at time $\alpha_t - \varepsilon$. Then, we get the following inequality.

$$\sum_{j<t} (r_{jt}^2 - d^2(j,i)) + \sum_{j \geq t} (\alpha_t - \rho d^2(j,i)) \leq \hat{f}. \tag{3}$$

This inequality follows from the fact that clients $j < t$ are already (directly) connected at the time $\alpha_t - \varepsilon$, and hence they bid $[r_{jt}^2 - d^2(j,i)]^+$ to i, and clients $j \geq t$ are still active, hence they bid $[\alpha - \rho d^2(j,i)]^+$ to i, where α is their current α value (which at time $\alpha_t - \varepsilon$, is $\alpha_t - \varepsilon$). We relax the terms of type $[x]^+$ to x, which can only make the LHS smaller. Taking $\varepsilon \to 0$ and since the algorithm ensures that at all times, every facility receives a total bid of at most $\hat{f}$, implying Equation (3).

Next, we eliminate r_{jt}^2 via triangle inequality. Since we know that at time $\alpha_t - \varepsilon$, t was not connected, we must have

$$\alpha_t \leq d^2(t, S) \leq (d(j,t) + r_{jt})^2 \tag{4}$$

See Lemma 4.9 for a formal proof. To get started with the spectral analysis, we will plug this lower bound on r_{jt} into Equation (3) and obtain a new inequality that upper bounds the sum of $\alpha_j - (\rho-1)R_j^2$ in terms of $\hat{f}$ and the sum of $d^2(j,i)$ (see Lemmas 4.11 and 4.12 for the formal statements we used for general and Euclidean metrics respectively).

We will use a slightly different approach to use the inequality $\alpha_t \leq (d(j,t) + r_{jt})^2$ for the metric case and the Euclidean case.

- Consider the Euclidean space of dimension dim. We will assume w.l.o.g. that the facility i is located at the origin of the Euclidean space and use the coordinate of the clients, $(x_j)_{j \in [s]} \in \mathbb{R}^{\mathsf{dim}}$, to rewrite $d(j,t) = \|x_j - x_t\|$ and $d(j,i) = \|x_j\|$. Using Equations (3) and (4), some quadratic inequalities and the Cauchy-Schwarz inequality (to make every $d(\cdot,\cdot)$ appear in a quadratic form in the inequalities), we can upper bound $\sum_j \alpha_j - (\rho-1)R_j^2$ by the sum of $\hat{f}$ and a quadratic form of $x = (x_j)_{j \in [s]} \in \mathbb{R}^{s \cdot \mathsf{dim}}$, denoted as $\hat{Q}(x)$. Because we have $\sum_j d^2(j,i) = \sum_j \|x_j\|^2 = \|x\|_2^2$, it suffices to upper bound $\frac{\hat{Q}(x)}{\|x\|_2^2}$. As an $(s \cdot \mathsf{dim}) \times (s \cdot \mathsf{dim})$ matrix, $\hat{Q}$ is a block-diagonal matrix with dim many identical $s \times s$ matrices on the diagonal, so an upper bound on the maximum eigenvalue of this $s \times s$ matrix yields a bound on the approximation ratio.

F: removed $\hat{f}+$ since we are talking about what we are adding to $\hat{f}$. Right?

- In the metric case, we use the metric triangle inequality $d(j,t) \leq d(j,i) + d(t,i)$. Then, we will consider each $d(j,i)$ as a variable a_j, and use Equations (3) and (4) and some quadratic inequalities to upper bound $\sum_j \alpha_j - (\rho-1)R_j^2$ by a sum of $\hat{f}$ and a quadratic form of $(a_j)_{j \in [s]}$, denoted as $\hat{Q}(a)$. Similarly, because we have $\sum_j d^2(j,i) = \sum_j a_j^2 = \|a\|^2$, it suffices to upper bound the maximum eigenvalue of $\hat{Q}(a)$ (again a quadratic form on s variables) to prove the bound on our objective Equation (2).

F: As before, removed $\hat{f}+$

Numerically, we found the following barriers for our approach:

- **Euclidean case:** After the Cauchy-Schwarz step (which makes the spectral analysis work), the underlying matrix $\hat{Q}(x)$ has an eigenvalue ≥ 3.51 in the case where there are 300 clients: the first 260 clients are immediately directly connected when they become inactive; while the last 40 clients are never directly connected.
- **Metric case**: The underlying matrix $\hat{Q}(x)$ has an eigenvalue ≥ 4.67 when the number of clients is large. More details can be found at the end of Subsection 2.3.

2.2 Spectral analysis for the Euclidean case

Let us fix $\rho = 2$. Recall that we are considering the special case where all the considered clients at any point of time are either active or directly connected. To further simplify the analysis, we present a weaker bound $\Gamma \leq 3 + \frac{27}{32} < 3.85$ for this case.

Let $x_1, x_2 \dots x_s$ be the vectors denoting the locations of the s clients, and suppose that the facility i is located at the all-zeros location, so that $d^2(j,i) = \|x_j\|^2$ for each $j \in [s]$. For simplicity, we will use $x = (x_j)_{j\in[s]}$ to denote the $s \cdot \mathsf{dim}$-dimensional vector by concatenating the s vectors. In the Euclidean case, we will rewrite Equation (4) by $\alpha_t \leq (r_{jt} + ||x_j - x_t||)^2$ for each $t > j$. Thus we get $\alpha_t \leq r_{jt}^2 + ||x_j - x_t||^2 + 2r_{jt}||x_j - x_t||$. Adding across all $t > j$ for a fixed j and using $r_{jt} \leq R_j$, we get

$$\sum_{t>j}(\alpha_t - r_{jt}^2) \leq \sum_{t>j} 2R_j||x_j - x_t|| + \sum_{t>j}||x_j - x_t||^2$$

Subtracting $s(\rho-1)R_j^2 = sR_j^2$,

$$\sum_{t>j}(\alpha_t - r_{jt}^2) - sR_j^2 \leq \sum_{t>j}\left(2R_j||x_j - x_t|| + ||x_j - x_t||^2\right) - sR_j^2$$

Applying the quadratic inequality $2ay - by^2 \leq \frac{a^2}{b}$, obtained from $\frac{1}{b}(a - by)^2 \geq 0$, we have

$$\sum_{t>j}(\alpha_t - r_{jt}^2) - sR_j^2 \leq \frac{\left(\sum_{t>j}||x_j - x_t||\right)^2}{s} + \sum_{t>j}||x_j - x_t||^2 \tag{5}$$

We sum Equation (3) over all $t \in [s]$ and the above Equation (5) over all $1 \leq j < t \leq s$, and then divide by s. We get

$$\sum_{t\in[s]} \alpha_t - R_t^2 \leq \hat{f} + Q(x) \ ,$$

where $Q(x)$ is a function of $x_1, x_2 \dots x_s$ alone:

$$Q(x) = \frac{1}{s}\left(\sum_j \frac{\left(\sum_{t>j}||x_j - x_t||\right)^2}{s} + \sum_j\sum_{t>j}||x_j - x_t||^2 + \sum_t\sum_{j<t}||x_j||^2 + \sum_t\sum_{j\geq t}\rho||x_j||^2\right) .$$

The first two terms come from the lower bound plugged in for r_{jt}^2 from above, and the last two terms come from the minus terms on the LHS of Equation (3).

Observe that the second term $\frac{1}{s}\sum_j \sum_{t>j} \|x_j - x_t\|^2$ equals to the optimal single-cluster cost for $[s]$ (see, e.g., Lemma B.3). Hence, it is no greater than $\sum_j \|x_j\|^2$. Rearranging the third and the fourth term via $\frac{1}{s}\sum_t \sum_j \rho\|x_j\|^2 - \frac{1}{s}\sum_t \sum_{j<t}(\rho-1)\|x_j\|^2$, we can rewrite them as $\rho\sum_j \|x_j\|^2 - (\rho-1)\sum_j \frac{s-j}{s}\|x_j\|^2$. Using our parameter $\rho = 2$, we have the following upper bound for $Q(x)$:

$$Q(x) \le \frac{1}{s^2}\sum_j \left(\sum\nolimits_{t>j} \|x_j - x_t\|\right)^2 - \sum_j \frac{s-j}{s}\|x_j\|^2 + 3\sum_j \|x_j\|^2.$$

Now we apply Cauchy-Schwarz on the first term to obtain $Q(x) \le \hat{Q}(x) + 3\sum_j \|x_j\|^2$, where

$$\hat{Q}(x) := \sum_j \frac{s-j}{s^2}\sum_{t>j} ||x_j - x_t||^2 - \sum_j \frac{s-j}{s}\|x_j\|^2 \ .$$

If we show that $\hat{Q}(x) \le \sum_j \lambda||x_j||^2$, then we obtain an LMP $(3+\lambda)$-approximation. This now reduces to upper bounding the largest eigenvalue of the matrix given by the quadratic form $\hat{Q}$. We next prove that $\lambda \le \frac{27}{32} < 0.85$.

F: As discussed, I dropped the part where we claim better λ: it is distracting and not critical

Consider two cases for j: $j \ge s/4$ and $j < s/4$. If $j \ge s/4$, we have $\frac{s-j}{s^2} \le \frac{3}{4s}$. The total contribution of j to $\hat{Q}(x)$ is then upper bounded by $\frac{3}{4s}\sum_{t>j}\|x_j - x_t\|^2$. If $j < s/4$, we have (in the second inequality, we use $(a-b)^2 \le 3a^2 + 1.5b^2$)

$$\begin{aligned}
\frac{s-j}{s^2}\sum_{t>j}\|x_j - x_t\|^2 - \frac{s-j}{s}\|x_j\|^2 &< \frac{1}{s}\sum_{t>j}\|x_j - x_t\|^2 - \frac{3}{4}\|x_j\|^2 \\
&\le \frac{3}{4s}\sum_{t>j}\|x_j - x_t\|^2 + \frac{1}{4s}\sum_{t>j}(3\|x_j\|^2 + 1.5\|x_t\|^2) - \frac{3}{4}\|x_j\|^2 \\
&\le \frac{3}{4s}\sum_{t>j}\|x_j - x_t\|^2 + \frac{3}{8s}\sum_{t>j}\|x_t\|^2
\end{aligned}$$

The intuition behind discussing these two different cases is as follows: for small j, there is a large subtraction term $-\frac{s-j}{s}\|x_j\|^2$; because of this, and the flexibility of squared triangle inequalities, we can let $\|x_j\|^2$ pay more for the cost in $\sum_{t>j}\|x_j - x_t\|^2$ (and let $\|x_t\|^2$ pay less).

Summing over all j, using the upper bound discussed above, we have

$$\begin{aligned}
\hat{Q}(x) &\le \frac{3}{4s}\sum_{j<t}\|x_j - x_t\|^2 + \frac{3}{8s}\sum_{j<s/4}\sum_{t>j}\|x_t\|^2 \\
&\le \frac{3}{4s}\sum_{j<t}\|x_j - x_t\|^2 + \frac{3}{32}\sum_t \|x_t\|^2 \\
&\le \frac{3}{4}\sum_j \|x_j\|^2 + \frac{3}{32}\sum_t \|x_t\|^2 = \frac{27}{32}\sum_j \|x_t\|^2 \ ,
\end{aligned}$$

where the last inequality follows from the fact that $\frac{1}{s}\sum_{j<t}\|x_j - x_t\|^2$ is the optimal single-cluster cost for $[s]$ among all possible centers in $\mathbb{R}^{\mathsf{dim}}$ while $\sum_j \|x_j\|^2$ is the cost of opening the center at the origin for $[s]$.

In Section 5, we generalize and refine the above derivation to the general case with both IC and DC clients to obtain the claimed LMP $(3+\ln 2)$-approximation for Facility Location in the Euclidean case. Our proof combines the above derivation with the techniques introduced in [CCG+26] to deal with clients in IC, and it is substantially more complex.

F: Dropped a sentence before that was redundant

F: Notice that LMP makes only sense for FL

2.3 Spectral analysis for the metric case

Now we present an overview of our spectral analysis for Metric k-Means in the special case when all clients are either directly connected or active (i.e., IC is empty at all times). We generalize it to all possible cases in Section 6.

Let $a_j = d(j,i)$. According to Equation (4) and the metric triangle inequality, we have $\alpha_t \le d^2(t,S_t) \le (r_{jt} + d(j,t))^2 \le (r_{jt} + d(j,i) + d(t,i))^2 = (r_{jt} + a_j + a_t)^2$. Thus, we can expand the square to get the upper bound $\alpha_t - r_{jt}^2 \le 2r_{jt}(a_j + a_t) + (a_j + a_t)^2$.

Fix any j. Adding across all $t > j$,

$$\begin{aligned}\sum_{t>j}(\alpha_t - r_{jt}^2) &\le \sum_{t>j} 2r_{jt}(a_j + a_t) + (a_j + a_t)^2 \\ &\le 2R_j\Big((s-j)a_j + \sum_{t>j} a_t\Big) + \sum_{t>j}(a_j + a_t)^2,\end{aligned}$$

where we crucially use the fact that each $r_{jt} \le R_j$ by the property of our algorithm. Now we subtract $s(\rho-1)R_j^2$ for j, and obtain

$$\begin{aligned}\sum_{t>j}(\alpha_t - r_{jt}^2) - s(\rho-1)R_j^2 &\le 2R_j\Big((s-j)a_j + \sum_{t>j} a_t\Big) - s(\rho-1)R_j^2 + \sum_{t>j}(a_j + a_t)^2 \\ &\le \frac{1}{s(\rho-1)}\Big((s-j)a_j + \sum_{t>j} a_t\Big)^2 + \sum_{t>j}(a_j + a_t)^2\end{aligned}$$

where the last inequality is obtained by observing that the quadratic terms are controlled by $2ax - bx^2 \le \frac{a^2}{b}$, which is equivalent to $\frac{1}{b}(a-bx)^2 \ge 0$.

If we now plug this lower bound for r_{jt}, and follow the same procedure of adding each inequality at time $\alpha_t - \varepsilon$ for $t \in [s]$ and divide by s, we would get a bound of the form $\sum_j(\alpha_j - (\rho-1)R_j^2) \le \hat{f} + \hat{Q}(a)$ where $\hat{Q}(a)$ is the new quadratic form on the distances arising from this process. If the maximum eigenvalue of the quadratic form corresponding to $\hat{Q}(a)$ is Γ, then this is at most $\hat{f} + \Gamma a^T a$, giving us an LMP Γ-approximation.

MC: Could point out that, in the general case, this is obtained by Lemma 6.2 in Section 6.1.

Now we can write the quadratic form $\hat{Q}(a)$ corresponding to our setup.

$$\hat{Q}(a) = \frac{1}{s}\left(\sum_j \frac{1}{s(\rho-1)}\Big((s-j)a_j + \sum_{t>j} a_t\Big)^2 + \sum_j\sum_{t>j}(a_j + a_t)^2 + \sum_t\sum_{j\ge t}\rho a_j^2 + \sum_t\sum_{j<t} a_j^2\right)$$

The first and second terms come from substituting the lower bounds for r_{jt} from the calculation above. The third and fourth terms come from the form of the bids of the clients at time $\alpha_t - \varepsilon$, clients $j \ge t$ are active and bid $\alpha_t - \rho a_j^2$ (hence the coefficient of ρ for a_j^2), whereas clients $j < t$ are directly connected and bid $r_{jt}^2 - a_j^2$ (hence the coefficient of 1 for a_j^2).

First, let us show via elementary analysis that when $\rho = 2$, we can bound $\Gamma \le 5$. The first term inside the parenthesis, $\sum_j \frac{1}{s}\big((s-j)a_j + \sum_{t>j} a_t\big)^2$, is equal to $\sum_j \frac{1}{s}\Big(\sum_{t>j}(a_j + a_t)\Big)^2$. Using the Cauchy-Schwarz inequality, this is at most $\sum_j \frac{s-j}{s}\sum_{t>j}(a_j + a_t)^2$, which is at most $\sum_j \frac{2(s-j)}{s}\sum_{t>j}(a_j^2 + a_t^2)$. Again, for the middle term, we get $\sum_j\sum_{t>j}(a_j + a_t)^2 \le \sum_j 2\sum_{t>j}(a_j^2 + a_t^2) = \sum_j 2\sum_{t>j}(a_j^2 + a_t^2)$.

Thus with $\rho = 2$, the expression is upper bounded by

$$\hat{Q}(a) \leq \frac{1}{s}\left(\sum_j \left(2 + \frac{2(s-j)}{s}\right)\sum_{t>j}(a_j^2 + a_t^2) + \sum_t \sum_{j\geq t} 2a_j^2 + \sum_t \sum_{j<t} a_j^2\right)$$

Now let us fix an index j^* and collect coefficients of $a_{j^*}^2$ for the term inside the parenthesis. The total coefficient is then

$$2(s-j^*)\left(1 + \frac{s-j^*}{s}\right) + 2\sum_{j<j^*}\left(1 + \frac{s-j}{s}\right) + 2j^* + (s-j^*) = 5s - j^* - 4 + \frac{j^*(j^*+1)}{s}$$

Dividing by the outer s and using $j^* \leq s$, the total coefficient is at most 5, which gives the desired inequality $\hat{Q}(a) \leq 5a^T a$, and hence $\Gamma \leq 5$.

However, it is possible to do better. By running experiments for various values of ρ, one can observe that the largest eigenvalue of the symmetric matrix describing this quadratic form is (numerically) minimized when $\rho \approx 2.5$. With this value of ρ, we numerically estimate the top eigenvalue to be 4.67783 as $s \to \infty$. It can be shown that the matrix corresponding to the quadratic form $\hat{Q}(a)$ is the sum of a diagonal and a rank one matrix, and hence it is not difficult to prove an analytic bound that reaches close to this constant for any s. Thus using our new algorithm with the new spectral analysis, we get an approximation ratio of 4.68 for this special case when no client is indirectly connected.

Thus all that is left is to generalize this to the case when there are indirectly connected clients. Let DC_t and IC_t be the sets of directly and indirectly connected clients at time $\alpha_t - \varepsilon$. Then the number of such valid sets[3] is exponential in s. Thus, we need a way to bound the general case directly.

For this purpose, we write down the equations for the most general IC/DC case, and try to solve them all at once, similarly to [CCG+25]. However, this analysis becomes more technically challenging as the matrices involved are no longer simple sums of diagonal and rank one matrices. Nevertheless, we show that the quadratic form in the general case can always be bounded by 4.9. Our proof uses the Collatz–Wielandt Formula which says that given a non-negative matrix M, if we find a non-zero vector h such that $\max_i \frac{(Mh)_i}{h_i}$ is at most Γ, then the top eigenvalue of M is at most Γ.

F: Slightly misleading since we get it only for a special case, so unfair to compare with 5.82. The worst case there is all DC, but I'm not sure we mention it (or wish to do that)

MC: 'all that is left is to generalize' is quite an understatement!

EL: @Ruiquan: I removed your sentence about the 2026 paper, because we did not show any difference between 25 and 26 so far at this point in the paper. If you want to highlight the differences, please add those comments in the appropriate places in Sections 5 and 6.

2.4 Turning LMP approximations to bicriteria approximations

To turn the LMP approximations to bi-criteria approximations (opening $k + O(\log n)$ centers), we employ the framework of [CGL+25; CCG+25; CCG+26]. The only technical improvement is as follows. The most recent work [CCG+26] can only turn an LMP Γ-approximation to a $\Gamma + \varepsilon$ bi-criteria approximations for any constant $\Gamma \geq 4$. We remove this condition, and now this step works for any constant $\Gamma > \rho$ (which is a clear lower bound for our LMP algorithm).

3 Preliminaries

For a client j and a set of facilities S, let $d(j, S) := \min_{i \in S} d(j, i)$. For any real number x, let $[x]_+ := \max\{x, 0\}$. For a positive integer n, we write $[n] := \{1, 2, \ldots, n\}$, and more generally,

AS: calling it a formula cause standard texts do so, and making a forward reference here. that's ok?

AA: I think one line to mention use of AI is alright here? TBD.

[3] The only condition for validity is that $DC_t \subseteq DC_{t+1}$ for all t, since the algorithm moves clients only from IC and A to DC, while never moving from DC to IC.

for integers $l \le r$, we write $[l:r] := \{l, l+1, \dots, r\}$. We use $\mathbb{R}_{>0}$ and $\mathbb{R}_{\ge 0}$ to denote the sets of positive and non-negative reals, respectively.

Facility Location with uniform opening costs and squared distances. In the *Facility Location with uniform opening costs and squared distances* problem, we are given a set of clients D, a set of facilities F, and a facility opening cost $f > 0$. The goal is to find a set $S \subseteq F$ minimizing

$$f|S| + \sum_{j \in D} d^2(j, S).$$

We write $\mathrm{OPT}_{\mathrm{FL}}(f)$ for the optimum value of this problem. We will refer to this problem simply as *Facility Location* throughout the paper.

AS: we don't really use OPT_{FL} anywhere

LP relaxation. The standard LP relaxation of Facility Location is

$$\begin{aligned} \min \quad & f \sum_{i \in F} y_i + \sum_{j \in D} \sum_{i \in F} d^2(j,i) x_{ij} & & \\ \text{s.t.} \quad & \sum_{i \in F} x_{ij} \ge 1 & & \forall j \in D, \\ & x_{ij} \le y_i & & \forall j \in D,\ i \in F, \\ & x_{ij} \ge 0,\ y_i \ge 0. & & \end{aligned} \tag{Primal FL LP}$$

Its dual can be written as

$$\begin{aligned} \max \quad & \sum_{j \in D} \alpha_j & & \\ \text{s.t.} \quad & \sum_{j \in D} [\alpha_j - d^2(j,i)]^+ \le f & & \forall i \in F, \\ & \alpha_j \ge 0 & & \forall j \in D. \end{aligned} \tag{Dual FL LP}$$

We write $\mathrm{OPT}_{\mathrm{LP}}(f)$ for the optimum value of the Dual FL LP.

Definition 3.1 (LMP approximation)**.** *Given $\Gamma \ge 1$, an algorithm is Lagrangian Multiplier Preserving (LMP) Γ-approximation for the Facility Location problem if, for every instance with facility opening cost $f > 0$, the output $S \subseteq F$ satisfies*

$$\Gamma f \cdot |S| + \sum_{j \in D} d^2(j, S) \ \le \ \Gamma \cdot \mathrm{OPT}_{\mathrm{LP}}(f).$$

Bounding distances. Thanks to standard reductions (see, e.g., [CCG+25]), in the metric case we will assume that distances are integers in $[1, n^3/\varepsilon]$ while losing a factor $1 + O(\varepsilon)$ in the approximation.

Lemma 3.2 ([CCG+25])**.** *For any constants $\varepsilon > 0$ and $\alpha > 1$, given a polynomial-time α-approximation algorithm for k-Means on instances with distances in $\{1, \dots, n^3/\varepsilon\}$, there exists a polynomial-time $\alpha(1 + O(\varepsilon))$-approximation algorithm for Metric k-Means.*

In the Euclidean case, we will assume that non-zero distances are in $[1, n^2/\varepsilon^2]$ thanks to the following reduction in [CCG+26].

F: Addressing issue 17

Lemma 3.3 ([CCG+26]). *For any given $\varepsilon > 0$ and $\alpha > 1$, given a polynomial-time α-approximation algorithm for Euclidean k-Means on instances with non-zero distances between 1 and n^2/ε^2, there exists a polynomial-time $\alpha(1 + O(\varepsilon))$-approximation algorithm for Euclidean k-Means on general instances.*

Triangle Inequalities for Squared Distances. We will use the following simple inequalities taken from [CCG+25]. Given three distances d, d_1, d_2 with $d \le d_1 + d_2$, the following lemma provides a useful upper bound on d^2 as a function of d_1 and d_2. The next lemma provides a similar upper bound in an analogous setting where $d \le d_1 + d_2 + d_3$.

Lemma 3.4 ([CCG+25]). *Let $\gamma > 1$ be any constant. For any x, y we have $\gamma x^2 + \frac{\gamma}{\gamma-1} y^2 \ge (x + y)^2$.*

Lemma 3.5 ([CCG+25]). *Let $\gamma > 1$ be any constant. For any x, y, z, we have $\gamma x^2 + \left(2 + \frac{2}{\gamma-1}\right)(y^2 + z^2) \ge (x + y + z)^2$.*

4 Our Improved Greedy Algorithm

In this section, we present our improved greedy LMP approximate algorithm for the Facility Location problem with uniform facility cost f. As in the previous works [CCG+25; CCG+26], our algorithm is based on the well-known greedy algorithm by Jain et al. [JMM+03].

In our algorithm, we use parameters $\rho, \Gamma > 1$, which will be defined (differently for the metric and Euclidean cases) later. We also use $\hat{f} = \Gamma \cdot f$, where f is the uniform facility opening cost. The following Algorithm 1 is our new greedy algorithm achieving the improved LMP approximation. As standard in the area, we present the algorithm in a continuous version where there is a variable θ (interpreted as *time*) that grows continuously from 0 to $+\infty$. Initially, all clients are active. We grow the α-values of active clients at the same speed as θ grows, until one of two conditions is met: (1) a new facility becomes openable, or (2) a client's α-value reaches its connection cost to some open facility. It is easy to discretize this algorithm. Here, we say one facility is openable if the total bid of the clients to this facility reaches $\hat{f}$, where the bid of client j to a facility i is defined as $[\alpha_j - \rho \cdot d^2(j, i)]^+$ when j is in A or IC, or as $[d^2(j, S) - d^2(j, i)]^+$ when j is in DC, at the time i is opened. In addition, for each client $j \in D$, we explicitly track its first direct-connection cost, R_j^2. We set $R_j = d(j, i^*)$ where i^* is the facility that caused j to be moved to DC for the first time. For a client that remains only indirectly connected or active, we keep $R_j = 0$ as it is initialized.

MC: Some repetition relative to Technical Overview, but ok.

MC: Algo 2 identical to Algo 1, correct?

RG: yes.

MC: So, can we make do with just one description of the algorithm?

MC: good to remind reader that S is the set of facilities opened so far.

Theorem 4.1. *Algorithm 1 is a LMP Γ-approximation for the Facility Location problem with uniform facility cost f if*

- *d is a (general) metric, $\rho = 2.5$ and $\Gamma = 4.9$; or*
- *d is a Euclidean metric, $\rho = 2$ and $\Gamma = 3 + \ln(2)$.*

Observation 4.2. *At any time during the algorithm, for any $j \in IC$, we have $\alpha_j = d^2(j, S)$.*

Because we freeze α_j when a client j becomes directly connected, we can show that the α values are non-increasing after connection. For some intermediate time θ, we will use $\alpha_j(\theta)$ to denote the α value of client $j \in D$ at that time. Further, for each $j \in D$, we define $\hat{\alpha}_j$ to be its α value at the first time when j is connected to some facility. Let $(\alpha_j^*)_{j \in D}$ be the final vector of α-values.

Algorithm 1 (Greedy Algorithm), restated.

Parameters: $\rho, \Gamma > 1$; $\hat{f} = \Gamma \cdot f$ where f is the uniform facility opening cost.

Initialization: $A = D$, $IC = DC = S = \emptyset$, and $R_j = \alpha_j = 0$ for every $j \in D$.

While $A \neq \emptyset$, uniformly grow the α_j of every client in A until:

1. For some unopened facility $i \in F \setminus S$, we have

$$\sum_{j \in A \cup IC} \left[\alpha_j - \rho \cdot d^2(j,i)\right]^+ + \sum_{j \in DC} \left[d^2(j,S) - d^2(j,i)\right]^+ = \hat{f}.$$

   In this case, we open facility i. For each client $j \in DC$, if $d^2(j,S) > d^2(j,i)$, we reconnect j to i. For each client $j \in A \cup IC$,

   - If $\alpha_j \geq \rho \cdot d^2(j,i)$, we (re)connect j to i, move j to DC and set $R_j = d(j,i)$.
   - Else, if $\alpha_j \geq d^2(j,i)$, we (re)connect j to i, set α_j to be $d^2(j,i)$, and move j to IC (if $j \in A$).

2. For some client $j \in A$, we have

$$\alpha_j = d^2(j,S).$$

   In this case, we move j to IC and connect j to the closest facility to it in S.

Observation 4.3. *For any client $j \in D$, $\alpha_j(\theta)$ is non-increasing in $\theta \in [\hat{\alpha}_j, +\infty)$.*

Proof. This follows from the fact that we only modify α_j after time $\hat{\alpha}_j$ when some facility i is opened and $\alpha_j > d^2(j,i)$. Then the resulting value of α_j is lowered to $d^2(j,i)$. □

Corollary 4.4. *For each client j, we have $\alpha_j^* \leq \hat{\alpha}_j$.*

Our goal is to show that $(\alpha_j^* - (\rho-1)R_j^2)_{j \in D}$ can characterize the total cost of Algorithm 1 and can be scaled to a feasible dual solution to the Dual FL LP.

First, we prove that the sum of $(\alpha_j^* - (\rho-1)R_j^2)_{j \in D}$ is an upper bound of the total cost. The proof relies on the following observation: for each client j that is finally directly connected, its α-value can be written as the sum of its final connection cost $d^2(j,S)$, $\rho - 1$ times its first direct-connection cost R_j^2, and the bids of j to the open facilities.

Lemma 4.5 (Dual Payment, Lemma 2.3 restated, replacing α by α^*)**.** *At the end of the execution of Algorithm 1,*

$$\sum_{j \in D} d^2(j,S) + |S|\hat{f} \leq \sum_{j \in D} (\alpha_j^* - (\rho-1)R_j^2).$$

Proof. First, observe that it is sufficient to prove that at any point in the execution, we have

$$\sum_{j \in DC} (\alpha_j - (\rho-1)R_j^2) \geq \sum_{j \in DC} d^2(j,S) + \sum_{i \in S} \hat{f}. \tag{6}$$

Indeed, at the end of the execution, $A = \emptyset$, and for every $j \in IC$, we have $\alpha_j^* = d^2(j, S)$ and $R_j = 0$ (since j never became directly connected). Thus, their contribution to the left-hand side is exactly their connection cost, making (6) imply the lemma.

To prove this, we proceed by induction and show that (6) is maintained throughout the execution.

The inequality is initially true since $DC = \emptyset$ and $S = \emptyset$. Furthermore, since whenever the α-value of a client j reaches $d^2(j, S)$ it stops growing and moves to IC, no client is added to DC except when a new facility is opened (i.e., added to S).

We now consider what happens when we open a facility i, i.e., add it to S. Let (α, S, A, IC, DC) be the state right before opening i, and set $DC' = \{j \in A \cup IC : \alpha_j \geq \rho d^2(j, i)\}$, and $X = \{j \in DC : d^2(j, S) > d^2(j, i)\}$. The change of cost of the right-hand side of (6) is exactly

$$\hat{f} + \sum_{j \in DC'} d^2(j, i) + \sum_{j \in X} (d^2(j, i) - d^2(j, S)) .$$

Since the algorithm decided to open i, we know that the total offer reaches $\hat{f}$, so we have

$$\hat{f} \leq \sum_{j \in DC'} (\alpha_j - \rho d^2(j, i)) + \sum_{j \in X} (d^2(j, S) - d^2(j, i)) .$$

Substituting this bound into our expression, we get that the change of cost of the right-hand side is at most

$$\sum_{j \in DC'} (\alpha_j - \rho d^2(j, i) + d^2(j, i)) = \sum_{j \in DC'} (\alpha_j - (\rho - 1) d^2(j, i)).$$

Because each client $j \in DC'$ is becoming directly connected for the first time, the algorithm sets its penalty variable $R_j = d(j, i)$, meaning $R_j^2 = d^2(j, i)$. Therefore, the change of the right-hand side is at most $\sum_{j \in DC'} (\alpha_j - (\rho - 1) R_j^2)$. Because we do not change the α-value or R_j value after a client is moved to DC, this is exactly the change of the left-hand side of (6). The invariant is thus maintained, completing the proof. □

Second, we prove that $\left((\alpha_j^* - (\rho - 1) R_j^2)/\Gamma\right)_{j \in D}$ can be scaled to a dual feasible solution for an appropriate choice of ρ, Γ. Since the proof of this part is very long, we will first present a general framework for both the metric case and Euclidean case in Section 4, where we establish a set of constraints for the variables $(\alpha_j^*)_{j \in D}$ and $(R_j)_{j \in D}$. Then, we will use the constraints to establish the following lemma for the metric case and the Euclidean case separately in Sections 5 and 6, using significantly different techniques.

Lemma 4.6 (Dual Feasibility, Lemma 2.4 restated, replacing α by α^*)**.** *For any facility $i \in F$, we have*

$$\sum_{j \in D} \left[\alpha_j^* - (\rho - 1) R_j^2 - \Gamma d^2(j, i)\right]^+ \leq \hat{f} ,$$

where

- *if d is a (general) metric, we choose $\rho = 2.5, \Gamma = 4.9$; or*
- *if d is a Euclidean metric, we choose $\rho = 2, \Gamma = 3 + \ln(2)$.*

AS: changing f to $\hat{f}$ in the Lemma 4.6

RG: This is a bug. I have fixed it.

MC: Note to standardize use of f vs $\hat{f}$ throughout.

Proof of Theorem 4.1. Consider the standard Dual FL LP for Facility Location problem with $f = \hat{f}/\Gamma$ and define

$$\widetilde{\alpha}_j \;:=\; \frac{\alpha_j^* - (\rho-1)R_j^2}{\Gamma} \qquad \forall j \in D.$$

By Lemma 4.6, $(\widetilde{\alpha}_j)_{j\in D}$ is dual feasible. Hence, by weak duality,

$$\sum_{j\in D} \widetilde{\alpha}_j \;\le\; \mathrm{OPT}_{\mathrm{LP}}(f)$$

Multiplying by Γ, we obtain

$$\sum_{j\in D} \big(\alpha_j^* - (\rho-1)R_j^2\big) \;\le\; \Gamma \cdot \mathrm{OPT}_{\mathrm{LP}}(f).$$

AS: we are defining LMP w.r.t the LP OPT and not the integral OPT. so changed.

On the other hand, Lemma 4.5 gives

$$\hat{f}\cdot|S| + \sum_{j\in D} d^2(j,S) \;\le\; \sum_{j\in D} \big(\alpha_j^* - (\rho-1)R_j^2\big).$$

Using the fact that $\hat{f} = \Gamma f$, we conclude that

$$\Gamma f \cdot |S| + \sum_{j\in D} d^2(j,S) \;\le\; \Gamma \cdot \mathrm{OPT}_{\mathrm{LP}}(f).$$

The claim follows from Definition 3.1. □

Dual feasibility: the analysis framework

Next, we show the analysis framework to establish Lemma 4.6. We will use $[n]$ to denote the set $\{1,2,\ldots,n\}$ and $[l:r]$ to denote the set $\{l, l+1, \ldots, r\}$. From now on, we fix any facility $i \in F$. Let $D^* = \{j \in D : \alpha_j^* - (\rho-1)R_j^2 > \Gamma \cdot d^2(j,i)\}$. Suppose that $|D^*| = s$ and we index them by $[s]$ and order them in the increasing order of the time they become inactive.

That is, we have $\hat{\alpha}_1 \le \hat{\alpha}_2 \le \cdots \le \hat{\alpha}_s$. Then, our goal (which will be formally shown at the end of this subsection) becomes proving

$$\sum_{j\in[s]} (\alpha_j^* - (\rho-1)R_j^2) \le \hat{f} + \Gamma \cdot \sum_{j\in[s]} d^2(j,i)\ . \tag{7}$$

Consider the time $\hat{\alpha}_t - \varepsilon$ for each client t, where $\varepsilon > 0$ is sufficiently small so that neither condition in the algorithm is satisfied between time $\hat{\alpha}_t - \varepsilon$ (inclusive) and $\hat{\alpha}_t$ (exclusive). For simplicity, we define DC^t as the set of directly connected clients in $[t-1]$ (i.e., $DC \cap [t-1]$) at time $\hat{\alpha}_t - \varepsilon$, and IC^t as the set of indirectly connected or active clients in $[t-1]$ (i.e., $(A \cup IC) \cap [t-1]$) at time $\hat{\alpha}_t - \varepsilon$. Moreover, we use S^t to denote the set S at time $\hat{\alpha}_t - \varepsilon$. Because the sets IC, DC, A are disjoint and the set DC only grows, we have the following observation.

Observation 4.7. *For each $t \in [s]$, we have $DC^t \cap IC^t = \emptyset$ and $DC^t \cup IC^t = [t-1]$. For each $t \in [s-1]$, we have $DC^t \subseteq DC^{t+1}$.*

Suppose that r_{jt} (for $t \in [s]$ and $j \in DC^t$) is the distance of client $j \in DC^t$ to its closest opened facility at time $\hat{\alpha}_t - \varepsilon$ (i.e., $d(j, S^t)$). One simple observation is that $r_{jt} \le R_j$ for each $j \in DC^t$, as the set S only grows in the algorithm.

Observation 4.8. *For each $t \in [s]$ and $j \in DC^t$, we have $r_{jt} \le R_j$.*

AS: i think you meant $s \ge t > j$

RG: I fixed the earlier definition.

Next, we will establish two sets of constraints for the variables $(\alpha_j^*)_{j\in[s]}$, $(r_{jt})_{1\le j<t\le s}$, and $(R_j)_{j\in[s]}$. The first set of constraints is as follows. The proof of the first set of constraints relies on the fact that for each client $t \in [s]$, its α-value at time $\hat{\alpha}_t - \varepsilon$ is strictly less than its distance to any open facility.

Lemma 4.9. *For each $t \in [s], j \in DC^t, \ell \in IC^t \cup \{t\}$, we have the following constraint:*

$$\alpha_\ell^* \le (r_{jt} + d(j,\ell))^2 \ . \tag{$p_{t,j,\ell}$}$$

Proof. Consider any client $t \in [s]$. Consider the time $\hat{\alpha}_t - \varepsilon$. Consider any client $\ell \in IC^t \cup \{t\}$. As $\ell \le t$, we have $\hat{\alpha}_\ell \le \hat{\alpha}_t$ by our assumption. Next, we prove that $\alpha_\ell^* \le d^2(\ell, S^t)$ by discussing two cases:

- If $\hat{\alpha}_\ell < \hat{\alpha}_t$ (since ε is sufficiently small, we have $\hat{\alpha}_\ell \le \hat{\alpha}_t - \varepsilon$), then ℓ is indirectly connected at time $\hat{\alpha}_t - \varepsilon$. Because of Observations 4.2 and 4.3, we have $\alpha_\ell^* \le \alpha_\ell(\hat{\alpha}_t - \varepsilon) = d^2(\ell, S^t)$.
- Otherwise (i.e., if $\hat{\alpha}_\ell = \hat{\alpha}_t$), we can prove $\hat{\alpha}_t - \varepsilon < d^2(\ell, S^t)$ by contradiction. Suppose that $\hat{\alpha}_t - \varepsilon \ge d^2(\ell, S^t)$. Then, ℓ will be connected to some opened facility using the second condition in Algorithm 1 at time $\hat{\alpha}_t - \varepsilon$, contradicting that ℓ is active at time $\hat{\alpha}_t - \varepsilon$. Since ε can be arbitrarily small here, we have $\hat{\alpha}_\ell = \hat{\alpha}_t \le d^2(\ell, S^t)$. Because of Corollary 4.4, we get $\alpha_\ell^* \le d^2(\ell, S^t)$.

Because the triangle inequality holds for the metric d, we have $d(\ell, S^t) \le d(j, S^t) + d(j, \ell)$. Therefore, $\alpha_\ell^* \le d^2(\ell, S^t) \le (r_{jt} + d(j,\ell))^2$, completing the proof. □

Our second set of constraints is as follows. The proof of the second set of constraints relies on the fact that for each client $t \in [s]$, the total bid to the fixed facility i at time $\hat{\alpha}_t - \varepsilon$ is strictly less than $\hat{f}$, as otherwise t will be connected to i at time $\hat{\alpha}_t - \varepsilon$ and become inactive, contradicting the definition of $\hat{\alpha}_t$.

Lemma 4.10. *For each $t \in [s]$, we have the following constraint:*

$$(s-t+1)\cdot\alpha_t^* + \sum_{\ell\in IC^t}\alpha_\ell^* + \sum_{j\in DC^t} r_{jt}^2 \le \hat{f} + \sum_{\ell\in IC^t\cup[t:s]} \rho\cdot d^2(\ell,i) + \sum_{j\in DC^t} d^2(j,i) \ . \tag{β_t}$$

Proof. Consider any client $t \in [s]$. Consider the time $\hat{\alpha}_t - \varepsilon$. Note that we assume that $\alpha_t^* > (\rho-1)R_t^2 + \Gamma\cdot d^2(t,i)$, where ρ, Γ are some constants strictly greater than 1. Because of Corollary 4.4 and because ε is sufficiently small, we have $\hat{\alpha}_t - \varepsilon \ge \alpha_t^* - \varepsilon > d^2(t,i)$. If facility i is opened at time $\hat{\alpha}_t - \varepsilon$, t should be connected to some facility at time $\hat{\alpha}_t - \varepsilon$, contradicting our assumption that $\hat{\alpha}_t$ denotes the time client t is connected for the first time. Therefore, the total bid to i at this time stamp should be strictly less than $\hat{f}$, i.e., (recall that $r_{jt} = d(j, S^t)$ for each $j \in DC^t$)

$$\sum_{j\in DC^t}\left[r_{jt}^2 - d^2(j,i)\right]^+ + \sum_{j\notin DC^t}\left[\alpha_j(\hat{\alpha}_t-\varepsilon) - \rho\cdot d^2(j,i)\right]^+ < \hat{f} \ .$$

If $j \ge t$, we have $\alpha_j(\hat{\alpha}_t - \varepsilon) = \hat{\alpha}_t - \varepsilon \ge \alpha_t^* - \varepsilon$. Otherwise, if $j < t$ and $j \notin DC^t$, we have either j being indirectly connected (i.e., $\hat{\alpha}_j \le \hat{\alpha}_t - \varepsilon$), or active (i.e., $\hat{\alpha}_j = \hat{\alpha}_t$ and $\alpha_j(\hat{\alpha}_t - \varepsilon) = \hat{\alpha}_t - \varepsilon$). If j is indirectly connected, because of Observation 4.3, we have $\alpha_j^* \le \alpha_j(\hat{\alpha}_t - \varepsilon)$. Otherwise,

because of Corollary 4.4, we have $\alpha_j^* - \varepsilon \le \hat{\alpha}_j - \varepsilon = \hat{\alpha}_t - \varepsilon = \alpha_j(\hat{\alpha}_t - \varepsilon)$. Because ε is sufficiently small, we have

$$\sum_{j \in DC^t} \left[r_{jt}^2 - d^2(j,i)\right]^+ + \sum_{j \in IC^t} \left[\alpha_j^* - \rho \cdot d^2(j,i)\right]^+ + \sum_{j \ge t} [\alpha_t^* - \rho \cdot d^2(j,i)]^+ \le \hat{f}.$$

Since $[x]^+ \ge x$, we can complete the proof by rearranging the terms in the above inequality. □

Finally, we only need to prove the following lemmas to complete the proof of Theorem 4.1. Because we will use the weighted version of Lemmas 4.11 and 4.12 in the log-adaptive algorithm and walking between two solutions later, we will directly claim the weighted generalization. To establish Lemma 4.6, we only need to consider the special case of them where $w(j) = 1$ for each $j \in [s]$.

RG: This might be more accurate

MC: Ensure that intro mentions walking.

AS: maybe mention here , why we need the weight function?

AS: is there a specific reason why in condition (3) we index the first two summations by ℓ, while last by j?

MC: Give this Lemma a name, e.g. Generalized Euclidean Dual Feasibility, or Weighted Euclidean Dual Feasibility? Could be helpful for the reader.

Lemma 4.11. *Let $\rho = 2$. Let $\hat{f}$ be any non-negative real number. Let $([s] \cup \{i\}, d)$ be a general metric. Let $w : [s] \to \mathbb{R}_{>0}$ be any positive weight function. Let $(IC^t)_{t \in [s]}$ and $(DC^t)_{t \in [s]}$ be two families of sets satisfying:*

(a) For each $t \in [s]$, $DC^t \cap IC^t = \emptyset$ and $DC^t \cup IC^t = [t-1]$.

(b) For each $t \in [s-1]$, $DC^t \subseteq DC^{t+1}$.

Suppose that the variables $(\alpha_j^*)_{j \in [s]}, (R_j)_{j \in [s]}, (r_{jt})_{1 \le j < t \le s}$ satisfy the following constraints

(1) For any $t \in [s], j \in DC^t$, we have $r_{jt} \le R_j$.

(2) For each $t \in [s], j \in DC^t$ and $\ell \in IC^t \cup \{t\}$, we have $\alpha_\ell^* \le (r_{jt} + d(j,\ell))^2$.

(3) For each $t \in [s]$, we have

$$\sum_{\ell \in [t:s]} w(\ell)\alpha_t^* + \sum_{\ell \in IC^t} w(\ell)\alpha_\ell^* + \sum_{j \in DC^t} w(j) r_{jt}^2 \le \hat{f} + \sum_{\ell \in IC^t \cup [t:s]} \rho \cdot w(\ell) d^2(\ell, i) + \sum_{j \in DC^t} w(j) d^2(j,i).$$

Then, the following holds:

$$\sum_{j \in [s]} w(j) \cdot (\alpha_j^* - (\rho - 1)R_j^2) \le \hat{f} + \Gamma \cdot \sum_{j \in [s]} w(j) \cdot d^2(j,i) \ , \tag{8}$$

where $\Gamma = 5$ if d is a general metric, and $\Gamma = 3 + \ln 2$ if d is a Euclidean metric.

Lemma 4.12. *Let $\rho = 2.5$ and $\Gamma = 4.9$. Let $\hat{f}$ be any non-negative real number. Let $([s] \cup \{i\}, d)$ be a metric. Let $w : [s] \to \mathbb{R}_{>0}$ be any positive weight function. Let $(IC^t)_{t \in [s]}$ and $(DC^t)_{t \in [s]}$ be two families of sets satisfying:*

(a) For each $t \in [s]$, $DC^t \cap IC^t = \emptyset$ and $DC^t \cup IC^t = [t-1]$.

(b) For each $t \in [s-1]$, $DC^t \subseteq DC^{t+1}$.

Suppose that the variables $(\alpha_j^*)_{j \in [s]}, (R_j)_{j \in [s]}, (r_{jt})_{t \in [s], j \in DC^t}$ satisfy the following constraints

(1) For any $t \in [s], j \in DC^t$, we have $r_{jt} \le R_j$.

(2) For each $t \in [s], j \in DC^t$, we have $\alpha_t^* \le (r_{jt} + d(j,t))^2$.

(3) For each $t \in [s]$, we have

$$\sum_{\ell \in [t:s]} w(\ell)\alpha_t^* + \sum_{\ell \in IC^t} w(\ell)\alpha_\ell^* + \sum_{j \in DC^t} w(j) r_{jt}^2 \le \hat{f} + \sum_{\ell \in IC^t \cup [t:s]} \rho w(\ell) d^2(\ell, i) + \sum_{j \in DC^t} w(j) d^2(j,i).$$

Then, the following holds:

$$\sum_{j\in[s]} w(j)\cdot(\alpha_j^* - (\rho-1)R_j^2) \le \hat{f} + \Gamma\cdot\sum_{j\in[s]} w(j)\cdot d^2(j,i)\ . \tag{9}$$

Remark 4.13. *The constraint (2) of Lemma 4.12 is a weaker version of the corresponding constraint in Lemma 4.11: we don't require the constraint for each $\ell \in IC^t \cup \{t\}$ (instead, we only require it for $\ell = t$).*

AS: using enumerate in Lemma 4.12

RG: fixed

Proof of Lemma 4.6. Observations 4.7 and 4.8 and Lemmas 4.9 and 4.10 imply that the variables $(\alpha_j^*)_{j\in[s]}, (R_j)_{j\in[s]}, (r_{jt})_{t\in[s],j\in DC^t}$ satisfy the constraints in Lemma 4.12 and Lemma 4.11. Therefore, we can apply Lemma 4.12 in the metric case and Lemma 4.11 in the Euclidean case to get the desired result. □

5 Establishing the LMP Approximations

RG: I will edit this section offline so keeping the current clean version on overleaf.

In this section, we will prove Lemma 4.11. For convenience, we restate the lemma below. Using the proof of Lemma 4.11, we will also prove a generalization of it, Lemma 5.20, which will be used in the later sections for the log-adaptive and robust analysis of the greedy algorithm for Euclidean metrics. Throughout this section, we will assume that $\rho = 2$. For simplicity, in the rest of this section, we will define $\Phi = \sum_{j\in[s]} w(j)\cdot d^2(j,i)$.

EL: EL: "special verison" sounds like a special case. Is it a generalization? Or just a variant?

Readers can assume $w(j) = 1$ for all $j \in [s]$ in the first reading, which is enough for understanding the main ideas and establishing the LMP approximations.

RG: I think it is now a generalization. I have changed the wording here.

Lemma 4.11. *Let $\rho = 2$. Let $\hat{f}$ be any non-negative real number. Let $([s]\cup\{i\}, d)$ be a general metric. Let $w : [s] \to \mathbb{R}_{>0}$ be any positive weight function. Let $(IC^t)_{t\in[s]}$ and $(DC^t)_{t\in[s]}$ be two families of sets satisfying:*

MC: Somewhere, we should explain why we need the generalization.

(a) For each $t \in [s]$, $DC^t \cap IC^t = \emptyset$ and $DC^t \cup IC^t = [t-1]$.

(b) For each $t \in [s-1]$, $DC^t \subseteq DC^{t+1}$.

Suppose that the variables $(\alpha_j^*)_{j\in[s]}, (R_j)_{j\in[s]}, (r_{jt})_{1\le j<t\le s}$ satisfy the following constraints

(1) For any $t \in [s], j \in DC^t$, we have $r_{jt} \le R_j$.

(2) For each $t \in [s], j \in DC^t$ and $\ell \in IC^t \cup \{t\}$, we have $\alpha_\ell^* \le (r_{jt} + d(j,\ell))^2$.

(3) For each $t \in [s]$, we have

$$\sum_{\ell\in[t:s]} w(\ell)\alpha_t^* + \sum_{\ell\in IC^t} w(\ell)\alpha_\ell^* + \sum_{j\in DC^t} w(j)r_{jt}^2 \le \hat{f} + \sum_{\ell\in IC^t\cup[t:s]} \rho\cdot w(\ell)d^2(\ell,i) + \sum_{j\in DC^t} w(j)d^2(j,i).$$

Then, the following holds:

$$\sum_{j\in[s]} w(j)\cdot(\alpha_j^* - (\rho-1)R_j^2) \le \hat{f} + \Gamma\cdot\sum_{j\in[s]} w(j)\cdot d^2(j,i)\ , \tag{8}$$

where $\Gamma = 5$ if d is a general metric, and $\Gamma = 3 + \ln 2$ if d is a Euclidean metric.

5.1 Proof Roadmap

To combine the constraints in the lemma, we will create a set of non-negative parameters $p_{tj\ell}$ for each $t \in [s], j \in DC^t, \ell \in IC^t \cup \{t\}$ such that $\sum_{\ell \in IC^t \cup \{t\}} p_{tj\ell} = 1$ for each $t \in [s]$ and each $j \in DC^t$. [4]

Using these parameters, we can lower bound the LHS of the third constraint (3) by adding the following quantity which is no greater than 0, according to the second constraint (2):

$$\begin{aligned} 0 &\geq \sum_{j \in DC^t} w(j) \cdot \sum_{\ell \in IC^t \cup \{t\}} p_{tj\ell}(\alpha^*_\ell - (r_{jt} + d(j,\ell))^2) \\ &= \sum_{j \in DC^t} w(j) \cdot \left(-r_{jt}^2 + \sum_{\ell \in IC^t \cup \{t\}} p_{tj\ell} \left(\alpha^*_\ell - 2r_{jt} \cdot d(j,\ell) - d^2(j,\ell) \right) \right) \end{aligned}$$

Therefore, we can lower bound the LHS of the third constraint (3) by the following quantity, where we use $r_{jt} \leq R_j$ in the second inequality:

$$\begin{aligned} &\text{LHS of constraint (3)} \\ &\geq \sum_{\ell \in [t:s]} w(\ell) \cdot \alpha^*_t + \sum_{\ell \in IC^t} w(\ell) \cdot \alpha^*_\ell + \sum_{j \in DC^t} w(j) \cdot \sum_{\ell \in IC^t \cup \{t\}} p_{tj\ell} \left(\alpha^*_\ell - 2r_{jt} \cdot d(j,\ell) - d^2(j,\ell) \right) \\ &\geq \sum_{\ell \in [t:s]} w(\ell) \cdot \alpha^*_t + \sum_{\ell \in IC^t} w(\ell) \cdot \alpha^*_\ell + \sum_{j \in DC^t} w(j) \cdot \sum_{\ell \in IC^t \cup \{t\}} p_{tj\ell} \left(\alpha^*_\ell - 2R_j \cdot d(j,\ell) - d^2(j,\ell) \right). \end{aligned}$$

Putting this inequality back to the third constraint (3), we have

$$\begin{aligned} &\sum_{\ell \in [t:s]} w(\ell)\alpha^*_t + \sum_{\ell \in IC^t} w(\ell)\alpha^*_\ell + \sum_{j \in DC^t} w(j) \sum_{\ell \in IC^t \cup \{t\}} p_{tj\ell}\alpha^*_\ell \\ \leq &\hat{f} + \rho \sum_{\ell \in IC^t \cup [t:s]} w(\ell)d^2(\ell,i) + \sum_{j \in DC^t} w(j)d^2(j,i) + \sum_{j \in DC^t} w(j)p_{tj\ell} \sum_{\ell \in IC^t \cup \{t\}} 2R_j d(j,\ell) + d^2(j,\ell) \\ = &\hat{f} + 2\Phi - \sum_{j \in DC^t} w(j)d^2(j,i) + \sum_{j \in DC^t} w(j)p_{tj\ell} \sum_{\ell \in IC^t \cup \{t\}} 2R_j d(j,\ell) + d^2(j,\ell) \qquad (\rho = 2) \end{aligned}$$

Further, we create another set of non-negative parameters β_t for each $t \in [s]$. [5] The goal is to ensure that $\sum_{t \in [s]} \beta_t \leq 1$ and the following inequality holds:

$$\sum_{t \in [s]} \beta_t \cdot \left(\sum_{\ell \in [t:s]} w(\ell)\alpha^*_t + \sum_{\ell \in IC^t} w(\ell)\alpha^*_\ell + \sum_{j \in DC^t} w(j) \sum_{\ell \in IC^t \cup \{t\}} p_{tj\ell}\alpha^*_\ell \right) = \sum_{j \in [s]} w(j)\alpha^*_j \,.$$

In this way, we can have the following upper bound:

$$\begin{aligned} &\sum_{j \in [s]} w(j) \left(\alpha^*_j - (\rho - 1)R_j^2 \right) \leq \hat{f} + 2\Phi \\ &\underbrace{- \sum_{j \in [s]} w(j)R_j^2 + \sum_{t \in [s]} \beta_t \cdot \left(\sum_{j \in DC^t} w(j)p_{tj\ell} \left(\sum_{\ell \in IC^t \cup \{t\}} 2R_j d(j,\ell) + d^2(j,\ell) \right) - w(j)d^2(j,i) \right)}_{\text{the extra cost } \mathsf{EC}}. \end{aligned} \tag{10}$$

[4] These parameters are corresponding to the parameters $\phi^{(t)}_{j,\ell}$ in [CCG+26]. We are using a slightly different notation here.

[5] These parameters are corresponding to the parameters β_t in [CCG+26].

To obtain Lemma 4.11, it suffices to explicitly define the parameters $(p_{tj\ell})_{t\in[s],j\in DC^t,\ell\in[s]}$, $(\beta_t)_{t\in IC^t\cup\{t\}}$ and prove the extra cost terms after $\hat{f}+2\Phi$ in the RHS of Equation (10) is no greater than $(\Gamma-\rho)\cdot\Phi$, where Γ depends on whether the underlying metric is metric or Euclidean according to Lemma 4.11.

5.2 Defining the parameters

In this subsection, we will define the parameters $(p_{tj\ell})_{t\in[s],j\in DC^t,\ell\in[s]}$ and $(\beta_t)_{t\in[s]}$ explicitly. These parameters are essentially the same as those in [CCG+26], but we write them in an explicit formula and prove some new properties here[6]. For the further establishment of the LMP approximation guarantee for this paper, we will also define $P_{j\ell}$, which indicates how much $d^2(j,\ell)$ will be charged in the extra cost term EC, and d_j^+, which indicates how much the extra cost is reduced by $d^2(j,i)$ (see Lemma 5.7 for a formal statement). After defining these parameters, we will prove Equation (8) in next subsection.

For simplicity, we will use $\mathsf{suf}(j)=\sum_{\ell\geq j}w(\ell)$ to denote the suffix sum of $w(j)$ for each $j\in[s]$. For each $t\in[s]$ and each $u\in[s]$, we consider

$$\pi_t(u)=\prod_{j\in DC^t:j<u}\frac{\mathsf{suf}(j)}{\mathsf{suf}(j+1)}\ ,$$

where if $[u-1]\cap DC^t=\emptyset$, we simply let $\pi_t(u)=1$. We define the following coefficients for each $t\in[s]$ and each $j\in[s]$:

$$\eta_{t\ell}=\begin{cases}\mathsf{suf}(t)\cdot\pi_t(t) & \text{if } \ell=t\ ,\\ w(\ell)\cdot\pi_t(\ell) & \text{if } \ell\in IC^t\ ,\\ 0 & \text{otherwise}\ ,\end{cases}\tag{11}$$

First, we present the **definitions of the parameters** $p_{tj\ell}$ for each $t\in[s]$, $j\in DC^t$, and $\ell\in IC^t\cup\{t\}$, so that $\sum_{\ell\in IC^t\cup\{t\}}p_{tj\ell}=1$ for each $t\in[s]$ and each $j\in DC^t$. For simplicity, we define $p_{tj\ell}=0$ if $\ell\notin IC^t\cup\{t\}$. The definition of $p_{tj\ell}$ will follow the rule below,

$$p_{tj\ell}\propto\begin{cases}0 & \text{if } \ell\leq j \text{ or } \ell\notin IC^t\cup\{t\}\ ,\\ \eta_{t\ell} & \text{if } \ell>j \text{ and } \ell\in IC^t\cup\{t\}\ .\end{cases}\tag{12}$$

Alternatively, we can define for each $t\in[s]$ and each $j\in DC^t$, the suffix sum of $\eta_{t\ell}$ as follows:

$$\sigma_t(j)=\sum_{\ell\geq j}\eta_{t\ell}\ ,\tag{13}$$

For any $t\in[s]$, $j\in DC^t$, and those $\ell\in[j+1:t]\cap(IC^t\cup\{t\})$, we can rewrite $p_{tj\ell}=\eta_{t\ell}/\sigma_t(j+1)$ (or, $\eta_{t\ell}/\sigma_t(j)$ because $\eta_{tj}=0$). For simplicity, we assume that $p_{tj\ell}=0$ if $j\notin DC^t$. It is clear that $\sum_{\ell\in IC^t\cup\{t\}}p_{tj\ell}=1$ for each $t\in[s]$ and each $j\in DC^t$. It is straightforward that we have $p_{tj\ell}\geq 0$. Further, we have the following properties:

Observation 5.1. *For each $t\in[s]$ and each $j\leq t$, we have $\sigma_t(j)=\mathsf{suf}(j)\cdot\pi_t(j)$. Moreover, if $j\in DC^t$, we have $\frac{w(j)}{\sigma_t(j)}=\frac{1}{\pi_t(j)}-\frac{1}{\pi_t(j+1)}$.*

[6]The definitions in [CCG+26] follow a recursion way. Properties here are either based on the explicit form or only implicitly proved in [CCG+26].

Proof. We first prove the first part by induction on j. For the base case ($j = t$), we have $\sigma_t(t) = \eta_{tt} = \mathsf{suf}(t) \cdot \pi_t(t)$. For the inductive step, we have $\sigma_t(j) = \eta_{tj} + \sigma_t(j+1) = \eta_{tj} + \mathsf{suf}(j+1) \cdot \pi_t(j+1)$. If $j \in DC^t$, we have $\pi_t(j) = \frac{\mathsf{suf}(j+1)}{\mathsf{suf}(j)} \cdot \pi_t(j+1)$. Because $\eta_{tj} = 0$, $\sigma_t(j) = \mathsf{suf}(j) \cdot \pi_t(j)$. Otherwise (if $j \notin DC^t$), we have $\pi_t(j) = \pi_t(j+1)$, and thus

$$\sigma_t(j) = \eta_{tj} + \mathsf{suf}(j+1)\pi_t(j) = w(j)\pi_t(j) + \mathsf{suf}(j+1)\pi_t(j) = \mathsf{suf}(j) \cdot \pi_t(j) \ .$$

For the second part, it is straightforward to establish the following equality:

$$\frac{1}{\pi_t(j)} - \frac{1}{\pi_t(j+1)} = \frac{1}{\pi_t(j)} - \frac{\mathsf{suf}(j+1)}{\mathsf{suf}(j)} \cdot \frac{1}{\pi_t(j)} = \frac{w(j)}{\mathsf{suf}(j+1) \cdot \pi_t(j)} = \frac{w(j)}{\sigma_t(j)} \ . \qquad \square$$

Corollary 5.2. *For each $t \in [s]$ and each $\ell \in IC^t \cup \{t\}$, we have $\sum_{j<\ell : j \in DC^t} \frac{w(j)}{\sigma_t(j)} = 1 - \frac{1}{\pi_t(\ell)}$.*

Lemma 5.3 (essentially, in the proof of [CCG+26])**.** *We have the following equality for any $(\alpha_j)_{j\in[s]}$ and any $t \in [s]$:*

$$\sum_{\ell \in [t:s]} w(\ell)\alpha^*_t + \sum_{\ell \in IC^t} w(\ell)\alpha^*_\ell + \sum_{j \in DC^t} w(j) \sum_{\ell \in IC^t \cup \{t\}} p_{tj\ell}\alpha^*_\ell = \sum_{j \in [s]} \eta_{tj}\alpha^*_j \ . \tag{14}$$

Proof. For each $t \in [s]$, we have

$$\begin{aligned}
&\sum_{\ell \in [t:s]} w(\ell)\alpha^*_t + \sum_{\ell \in IC^t} w(\ell)\alpha^*_\ell + \sum_{j \in DC^t} w(j) \sum_{\ell \in IC^t \cup \{t\}} p_{tj\ell}\alpha^*_\ell \\
&= \mathsf{suf}(t)\alpha^*_t + \sum_{\ell \in IC^t} w(\ell)\alpha^*_\ell + \sum_{j \in DC^t} w(j) \sum_{\ell \in IC^t \cup \{t\} : \ell > j} \frac{\eta_{t\ell}}{\sigma_t(j)} \cdot \alpha^*_\ell \\
&= \mathsf{suf}(t)\alpha^*_t + \sum_{\ell \in IC^t} w(\ell)\alpha^*_\ell + \sum_{\ell \in IC^t \cup \{t\}} \alpha^*_\ell \cdot \eta_{t\ell} \cdot \sum_{j \in DC^t : j < \ell} \frac{w(j)}{\sigma_t(j)} \\
&= \mathsf{suf}(t)\alpha^*_t + \sum_{\ell \in IC^t} w(\ell)\alpha^*_\ell + \sum_{\ell \in IC^t \cup \{t\}} \alpha^*_\ell \cdot \eta_{t\ell} \cdot \left(1 - \frac{1}{\pi_t(\ell)}\right) && \text{(Corollary 5.2)} \\
&= \sum_{\ell \in IC^t \cup \{t\}} \eta_{t\ell}\alpha^*_\ell + \mathsf{suf}(t)\alpha^*_t + \sum_{\ell \in IC^t} w(\ell)\alpha^*_\ell - \sum_{\ell \in IC^t \cup \{t\}} \frac{\eta_{t\ell}}{\pi_t(\ell)} \cdot \alpha^*_\ell \\
&= \sum_{\ell \in IC^t \cup \{t\}} \eta_{t\ell}\alpha^*_\ell \ . && \text{(by definition of } \eta_{t\ell} \text{ in Equation (11))}
\end{aligned}$$

Because we have $\eta_{t\ell} = 0$ if $\ell \notin IC^t \cup \{t\}$, we can rewrite the LHS of Equation (14) as $\sum_{t\in[s]} \beta_t \cdot \sum_{\ell \in [s]} \eta_{t\ell}\alpha^*_\ell$. $\square$

Next, we present **the definitions of the parameters** β_t for each $t \in [s]$ by the following recurrence relation (note that $\eta_{jj} > 0$ for any $j \in [s]$):

$$\beta_j \cdot \eta_{jj} = w(j) - \sum_{t>j} \beta_t \cdot \eta_{tj}. \tag{15}$$

The definitions are in the same spirit as the definitions of the parameters β_t in the metric case (Lemma 6.1). Note that because we have $\eta_{tj} = 0$ for any $t < j$, we have

Observation 5.4. *For each $j \in [s]$, we have $\sum_{t\in[s]} \beta_t \eta_{tj} = w(j)$.*

We will defer the proof that $\beta_j \geq 0$ to Lemma 5.16 and the proof that $\sum_{t\in[s]} \beta_t \leq 1$ to Corollary 5.18. Using the above definition and Lemma 5.3, we can show that

Corollary 5.5. *We have the following equality for any $(\alpha_j)_{j\in[s]}$:*

$$\sum_{t\in[s]} \beta_t \cdot \left(\sum_{\ell\in[t:s]} w(\ell)\alpha^*_t + \sum_{\ell\in IC^t} w(\ell)\alpha^*_\ell + \sum_{j\in DC^t} w(j) \sum_{\ell\in IC^t\cup\{t\}} p_{tj\ell}\alpha^*_\ell \right) = \sum_{j\in[s]} w(j)\alpha^*_j \ . \quad (16)$$

Finally, for simplicity, we **construct the matrix** P, in which, for any $j < \ell$, $P_{j\ell}$ equals the coefficient (divided by $2w(j)$) of $d(j,\ell)$ in the extra cost term EC in Equation (10). We define $P_{j\ell} = 0$ for any $j \geq \ell$. Intuitively, we can think of $P_{j\ell}$ as how much we will charge $d^2(j,\ell)$ in the extra cost term EC (see Lemma 5.7 for a formal statement). Because we have $p_{tj\ell} \neq 0$ only when $j \in DC^t$, $\ell \in IC^t \cup \{t\}$ and $j < \ell$, we have following simple expression for $P_{j\ell}$:

$$P_{j\ell} = \sum_{t\in[s]} \beta_t \cdot p_{tj\ell} \ .$$

We also **define** $d^+_j = \sum_{\ell>j} P_{j\ell}$. Intuitively, this indicates how much the extra cost is reduced by $d^2(j,i)$ (see Lemma 5.7 for a formal statement).

Further, we present a simple characterization of d^+_j in terms of $\{\beta_t\}$, which will be used in the next subsection to upper bound the extra cost term EC in Equation (10) by Φ. We define $t_j \geq j$ as the threshold such that for any $t > t_j$, we have $j \in DC^t$, and for any $j < t \leq t_j$, we have $j \notin DC^t$.

Observation 5.6. *For each $j \in [s]$, we have $d^+_j = \sum_{t:j\in DC^t} \beta_t = \sum_{t>t_j} \beta_t$.*

Proof. By definition, we have

$$d^+_j = \sum_{\ell>j} P_{j\ell} = \sum_{\ell>j}\sum_{t\in[s]} \beta_t \cdot p_{tj\ell} = \sum_{t\in[s]:j\in DC^t} \beta_t \cdot \sum_{\ell>j} p_{tj\ell} = \sum_{t>t_j} \beta_t \ . \qquad \square$$

5.3 Proof of Lemma 4.11

5.3.1 Step 1: Bounding the extra cost term via $P_{j\ell}$ and d^+_j

Using above definitions, we can rewrite the extra cost term in Equation (10):

Lemma 5.7. *We have the following upper bound for the extra cost term EC in Equation (10):*

$$\mathsf{EC} \leq \sum_{j<\ell} w(j)P_{j\ell}d^2(j,\ell) + \sum_{j\in[s]} w(j)d^+_j \sum_{\ell>j} P_{j\ell}d^2(j,\ell) - \sum_{j\in[s]} w(j)d^+_j d^2(j,i) \ .$$

Proof. According to our definition of $P_{j\ell}$ and d^+_j, we can rewrite the extra cost term EC in

Equation (10) as follows:

$$\begin{aligned}
\mathsf{EC} &\le -\sum_{j\in[s]} w(j)R_j^2 + \sum_{j<\ell} w(j)P_{j\ell}\cdot\left(2R_j\cdot d(j,\ell)+d^2(j,\ell)\right) - \sum_{j\in[s]} w(j)d_j^+\cdot d^2(j,i)\\
&\le \sum_{j<\ell} w(j)P_{j\ell}d^2(j,\ell) + \sum_{j\in[s]} w(j)\left(-R_j^2 + \sum_{\ell>j} P_{j\ell}\cdot 2R_j d(j,\ell)\right) - \sum_{j\in[s]} w(j)d_j^+ d^2(j,i)\\
&\le \sum_{j<\ell} w(j)P_{j\ell}d^2(j,\ell) + \sum_{j\in[s]} w(j)\left(\sum_{\ell>j} P_{j\ell}d(j,\ell)\right)^2 - \sum_{j\in[s]} w(j)d_j^+ d^2(j,i)\\
&\qquad\qquad\text{(Applying } -R_j^2+2bR_j\le b^2 \text{ for any } b\in\mathbb{R})\\
&\le \sum_{j<\ell} w(j)P_{j\ell}d^2(j,\ell) + \sum_{j\in[s]} w(j)d_j^+\sum_{\ell>j} P_{j\ell}d^2(j,\ell) - \sum_{j\in[s]} w(j)d_j^+ d^2(j,i)\\
&\qquad\qquad\text{(Cauchy-Schwarz inequality)}
\end{aligned}$$

□

5.3.2 Step 2: Bounding the Laplacian matrix of P

The following lemma upper bounds the Laplacian matrix of P by the identity matrix, which is essentially a key lemma in [CCG+26] to establish the LMP 4-approximation for Euclidean k-means. For the sake of completeness, we prove this lemma in Subsection B.1.

Lemma 5.8 (essentially, [CCG+26])**.** *For any $w:[s]\to\mathbb{R}_{>0}$ and any $x_1,\dots,x_s\in\mathbb{R}$, we have*

$$\sum_{j<\ell} w(j)P_{j\ell}(x_j-x_\ell)^2 \le \sum_{j\in[s]} w(j)x_j^2\ .$$

As a corollary, we have the following properties of the matrix P, which give bounds for the first term of Lemma 5.7.

Corollary 5.9. *For any metric $([s]\cup\{i\},d)$, we have*

$$\sum_{j<\ell} w(j)P_{j\ell}d^2(j,\ell) \le 2\sum_{j\in[s]} w(j)d^2(j,i)\ . \tag{17}$$

In particular, if d is Euclidean, we have an improved upper bound:

$$\sum_{j<\ell} w(j)P_{j\ell}d^2(j,\ell) \le \sum_{j\in[s]} w(j)d^2(j,i)\ .$$

Proof. Let $a_j=d(j,i)$ for each $j\in[s]$. By the triangle inequality, we have $d(j,\ell)\le a_j+a_\ell$ for any $j<\ell$. Therefore, we have

$$\sum_{j<\ell} w(j)P_{j\ell}d^2(j,\ell) \le \sum_{j<\ell} w(j)P_{j\ell}(a_j+a_\ell)^2 \le 2\sum_{j<\ell} w(j)P_{j\ell}(a_j^2+a_\ell^2)\ .$$

For each $j\in[s]$, consider the vector $x=(x_1,\dots,x_s)$ such that $x_j=a_j$ and $x_\ell=0$ for all $\ell\ne j$. By Lemma 5.8, we have $\sum_{\ell<j} w(\ell)P_{\ell j}a_j^2+\sum_{\ell>j} w(j)P_{j\ell}a_j^2\le w(j)a_j^2$. Therefore, we have obtained the desired Equation (17).

In particular, if d is Euclidean, we can w.l.o.g. assume that i is located at the origin 0^{dim} and j is located at $v_j \in \mathbb{R}^{\mathsf{dim}}$ for each $j \in [s]$. Then we can rewrite $\sum_{j<\ell} w(j)P_{j\ell}d^2(j,\ell)$ as $\sum_{t\in[\mathsf{dim}]}\sum_{j<\ell} w(j)P_{j\ell}(v_{jt}-v_{\ell t})^2$. By Lemma 5.8, we have $\sum_{j<\ell} w(j)P_{j\ell}(v_{jt}-v_{\ell t})^2 \le \sum_{j\in[s]} w(j)v_{jt}^2$ for each $t \in [\mathsf{dim}]$. Therefore, we have

$$\sum_{j<\ell} w(j)P_{j\ell}d^2(j,\ell) \le \sum_{t\in[\mathsf{dim}]}\sum_{j\in[s]} w(j)v_{jt}^2 = \sum_{j\in[s]} w(j)d^2(j,i)\ .$$

Hence we have completed the proof. □

Another direct corollary of this lemma is that the sum of P on any column ℓ is upper bounded by $w(\ell)(1-d_\ell^+)$.

Corollary 5.10. *For any $\ell \in [s]$, we have $\sum_{j<\ell} w(j)P_{j\ell} \le w(\ell)(1-d_\ell^+)$.*

Proof. Consider the vector $x = e_\ell$ which is the indicator vector for ℓ. By Lemma 5.8, we have $\sum_{j<\ell} w(j)P_{j\ell} + \sum_{\ell<\ell'} w(\ell)P_{\ell\ell'} \le w(\ell)$. Because $d_\ell^+ = \sum_{\ell'>\ell} P_{\ell\ell'}$, we have $\sum_{j<\ell} w(j)P_{j\ell} \le w(\ell)(1-d_\ell^+)$. □

5.3.3 Step 3: A key property on the conditional column sum of P

The third key property is an upper bound for the conditional column sum of P in terms of d_j^+ and $w(\ell)$, which will be used to upper bound the second and the third term in Lemma 5.7. Intuitively, consider the ℓ-th column (corresponding to client ℓ) in the matrix P. The conditional column sum of P is the sum of all rows j in this column whose corresponding client j has $d_j^+ > \theta$. For a better understanding, d_j^+ can be interpreted as a parameter indicating whether client j becomes directly connected early in the process of the greedy algorithm (according to Observation 5.6). If $d_j^+ > d_{j'}^+$ for other client j', it means that client j becomes directly connected earlier than j'. Hence, the conditional column sum of P can be interpreted as the sum of the coefficients of a set of clients $j < \ell$ that become directly connected at some time point before ℓ becomes directly connected in the process of the greedy algorithm. The reason we only need to take care of those clients j is because we only charge $d^2(j,\ell)$ for those clients j in the upper bound of the extra cost term EC in Lemma 5.7 (i.e., $P_{j'\ell} = 0$ for other clients j').

Lemma 5.11. *For any $\theta \in (0,1)$ and any $\ell \in [s]$, we have*

$$\sum_{j<\ell} \mathbb{1}(d_j^+ > \theta)\cdot w(j)P_{j\ell} \le w(\ell)(1-\theta).$$

Remark 5.12. *We shall remark that if $\theta < d_\ell^+$, this lemma has a weaker bound than Corollary 5.10. However, if $\theta \ge d_\ell^+$, this lemma gives a stronger bound than Corollary 5.10.*

To prove this lemma, we start with the following simple observation, which is a direct consequence of the definition of $\pi_t(j)$.

Observation 5.13. *For any $j \in [s]$, $\pi_t(j)$ is non-decreasing in t.*

Proof. This observation follows the definition of $\pi_t(j)$ and the fact that $DC^t \subseteq DC^{t+1}$ for any $t \in [k-1]$. □

Corollary 5.14. *Suppose $u \le t$. Then, for any $\ell \in [s]$, we have*

$$\sum_{j<\ell : j \in DC^u} \frac{w(j)}{\sigma_t(j)} \le \sum_{j<\ell : j \in DC^u} \frac{w(j)}{\sigma_u(j)} \ .$$

Proof. According to Observation 5.1, we have $\sigma_t(j) = \mathsf{suf}(j) \cdot \pi_t(j)$ and $\sigma_u(j) = \mathsf{suf}(j) \cdot \pi_u(j)$. Because $\pi_t(j)$ is non-decreasing in t, we have $\pi_t(j) \ge \pi_u(j)$, and thus $\frac{w(j)}{\sigma_t(j)} \le \frac{w(j)}{\sigma_u(j)}$ for each $j \in DC^u$. Therefore, we have $\sum_{j<\ell : j \in DC^u} \frac{w(j)}{\sigma_t(j)} \le \sum_{j<\ell : j \in DC^u} \frac{w(j)}{\sigma_u(j)}$. □

Our first intermediate lemma is an upper bound for the conditional column sum of P restricted to a subset of clients DC^u for any $u \in [s]$.

Lemma 5.15. *Fix any $u \in [s]$. For any $\ell \in [s]$, we have*

$$\sum_{j<\ell : j \in DC^u} w(j) P_{j\ell} \le w(\ell) \cdot \left(1 - \frac{1}{\pi_u(\ell)}\right) \ .$$

Proof. Using the definition of $P_{j\ell}$, we have

$$\sum_{j<\ell : j \in DC^u} w(j) P_{j\ell} = \sum_{j<\ell : j \in DC^u} w(j) \sum_{t \in [s]} \beta_t \cdot p_{tj\ell} = \sum_{t \in [s]} \beta_t \cdot \sum_{j<\ell : j \in DC^u} w(j) p_{tj\ell} \ .$$

First, we consider the case when $t < u$. In this case, we have $DC^t \subseteq DC^u$. Since by our definition, $p_{tj\ell} = 0$ if $j \notin DC^t$, we have $\sum_{j<\ell : j \in DC^u} w(j) p_{tj\ell} = \sum_{j<\ell : j \in DC^t} w(j) p_{tj\ell}$. Therefore, for $t < u$, we have the following equation, where the second equality is by our definition of $p_{tj\ell}$ and the final equality is by Corollary 5.2:

$$\sum_{j<\ell : j \in DC^u} w(j) p_{tj\ell} = \sum_{j<\ell : j \in DC^t} w(j) p_{tj\ell} = \sum_{j<\ell : j \in DC^t} w(j) \frac{\eta_{t\ell}}{\sigma_t(j)} = \eta_{t\ell} \left(1 - \frac{1}{\pi_t(\ell)}\right) \ .$$

Since $\pi_t(\ell)$ is non-decreasing in t, we have $\sum_{j<\ell : j \in DC^u} w(j) p_{tj\ell} \le \eta_{t\ell}(1 - \frac{1}{\pi_u(\ell)})$ for $t < u$.

Second, we consider the case when $t \ge u$. Using our definition of $p_{tj\ell}$ and Corollary 5.2, we have

$$\begin{aligned} \sum_{j<\ell : j \in DC^u} w(j) p_{tj\ell} = \sum_{j<\ell : j \in DC^u} w(j) \cdot \frac{\eta_{t\ell}}{\sigma_t(j)} = \eta_{t\ell} \sum_{j<\ell : j \in DC^u} \frac{w(j)}{\sigma_t(j)} &\le \eta_{t\ell} \sum_{j<\ell : j \in DC^u} \frac{w(j)}{\sigma_u(j)} \\ &= \eta_{t\ell} \left(1 - \frac{1}{\pi_u(\ell)}\right) \ . \end{aligned}$$

In conclusion, using the above discussions, the recurrence relation for β_t (Equation (15)) and the fact that $\eta_{t\ell} = 0$ for $t < \ell$, we have

$$\sum_{j<\ell : j \in DC^u} w(j) P_{j\ell} = \sum_{t \in [s]} \beta_t \sum_{j<\ell : j \in DC^u} w(j) p_{tj\ell} \le \sum_{t \in [s]} \beta_t \eta_{t\ell} \left(1 - \frac{1}{\pi_u(\ell)}\right) = w(\ell) \left(1 - \frac{1}{\pi_u(\ell)}\right) .$$

□

Next, we show the first deferred proof that $\beta_u \ge 0$ for any $u \in [s]$.

Lemma 5.16. *For any $u \in [s]$, we have $\beta_u \ge 0$.*

Proof. First, we show an upper bound for the following term:

$$\begin{aligned}
\sum_{t>u} \beta_t \cdot \eta_{tu} \le \sum_{t>u} \beta_t \cdot w(u)\pi_t(u) &= \sum_{t>u} \beta_t \cdot \frac{w(u)\sigma_t(u)}{\mathsf{suf}(u)} && \text{(Observation 5.1)} \\
&= \frac{w(u)}{\mathsf{suf}(u)} \cdot \sum_{t>u} \beta_t \cdot \sum_{j\ge u} \eta_{tj} \\
&= \frac{w(u)}{\mathsf{suf}(u)} \cdot \sum_{j\ge u} \sum_{t>u} \beta_t \cdot \eta_{tj} \\
&= \frac{w(u)}{\mathsf{suf}(u)} \cdot \left(\sum_{j>u} w(j) + \sum_{t>u} \beta_t \eta_{tu} \right) && \text{(Equation (15))} \\
&= w(u) \cdot \frac{\mathsf{suf}(u+1)}{\mathsf{suf}(u)} + \frac{w(u)}{\mathsf{suf}(u)} \cdot \sum_{t>u} \beta_t \cdot \eta_{tu}.
\end{aligned}$$

Therefore, $\frac{\mathsf{suf}(u+1)}{\mathsf{suf}(u)} \sum_{t>u} \beta_t \cdot \eta_{tu} \le w(u) \cdot \frac{\mathsf{suf}(u+1)}{\mathsf{suf}(u)}$, and thus $\sum_{t>u} \beta_t \cdot \eta_{tu} \le w(u)$.

Recall Equation (15), where we have $\beta_u \eta_{uu} = w(u) - \sum_{t>u} \beta_t \cdot \eta_{tu}$. Because $\eta_{uu} > 0$, we have $\beta_u \ge 0$. □

AS: i think a typo here in Equation (15), it should be η_{tu} not τ_{tu}/

RG: fixed.

We show a lemma that could imply the second deferred proof that $\sum_{t\in[s]} \beta_t \le 1$. This lemma is more general in the sense that it also gives an upper bound for the tail sum of β_t, which is useful for establishing Lemma 5.11, the focus of this part.

Lemma 5.17. *For any $u \in [s]$, we have $\sum_{t\ge u} \beta_t \le \frac{1}{\pi_u(u)}$.*

AS: super minor, but for consistency i think we should either index all lemmas by either $[s]$ or $[s]$.

RG: fixed

Proof. For this lemma, we will prove a stronger statement that for any $u \in [s]$, we have

$$\sum_{t\ge u} \beta_t \cdot \pi_t(u) = 1 \ . \tag{18}$$

Then, because $\pi_t(u)$ is non-decreasing in t, we have $\sum_{t\ge u} \beta_t \le \frac{1}{\pi_u(u)}$.

To prove the stronger statement, we will use Observation 5.1:

$$\begin{aligned}
\sum_{t\ge u} \beta_t \cdot \pi_t(u) &= \sum_{t\ge u} \beta_t \cdot \frac{\sigma_t(u)}{\mathsf{suf}(u)} && \text{(Observation 5.1)} \\
&= \frac{1}{\mathsf{suf}(u)} \cdot \sum_{t\ge u} \beta_t \cdot \sum_{j\ge u} \eta_{tj} \\
&= \frac{1}{\mathsf{suf}(u)} \cdot \sum_{j\ge u} \sum_{t\ge u} \beta_t \cdot \eta_{tj} \\
&= \frac{1}{\mathsf{suf}(u)} \cdot \sum_{j\ge u} w(j) = 1. && \text{(by Equation (15))}
\end{aligned}$$

□

By choosing $u = 1$ for Equation (18) in the proof, we have the following corollary as we have $\pi_t(u) = 1$ for $t \in [s]$.

Corollary 5.18. *We have $\sum_{t\in[s]} \beta_t = 1$.*

This further implies the following corollary because of Observation 5.6.

Corollary 5.19. *For each $j \in [s]$, we have $d_j^+ \le 1$.*

Finally, we are ready to present the proof of the lemma in this part.

Proof of Lemma 5.11. Suppose that u^* is the largest index u such that $\sum_{t \ge u} \beta_t > \theta$. Such index u always exists because of Corollary 5.18. Then, according to Observation 5.6, we have $d_j^+ > \theta$ if and only if $t_j < u^*$, i.e., $j \in DC^{u^*}$. Therefore, we have

$$\begin{aligned}
\sum_{j<\ell} \mathbb{1}(d_j^+ > \theta) \cdot w(j) P_{j\ell} &= \sum_{j<\ell : j \in DC^{u^*}} w(j) P_{j\ell} \\
&\le w(\ell) \left(1 - \frac{1}{\pi_{u^*}(\ell)}\right) && \text{(Lemma 5.15)} \\
&\le w(\ell) \left(1 - \frac{1}{\pi_{u^*}(u^*)}\right) \\
&\le w(\ell) \left(1 - \sum_{t \ge u^*} \beta_t\right) < w(\ell)(1-\theta) \ ,
\end{aligned}$$

where the second inequality is using two facts: (i) $\pi_{u^*}(\ell)$ is non-decreasing in ℓ, so $\pi_{u^*}(\ell) \le \pi_{u^*}(u^*)$ for $\ell \le u^*$; and (ii) we have $DC^{u^*} \subseteq [u^* - 1]$ and so $\pi_{u^*}(\ell) = \pi_{u^*}(u^*)$ for $\ell \ge u^*$. □

5.3.4 Step 4(a): Proof for the Euclidean case

In this part, we will prove Lemma 4.11 for the Euclidean case. We will consider $\Gamma = 3 + \ln 2$ here. Without loss of generality, we assume the facility i is located at $x_i = 0^{\mathsf{dim}}$ and the corresponding s clients are located at $x_1, \ldots, x_s$ respectively. According to Equation (10) and Lemma 5.7, we have

$$\begin{aligned}
&\sum_{j \in [s]} w(j) \left(\alpha_j^* - (\rho - 1) R_j^2\right) \\
&\quad \le \hat{f} + 2\Phi + \sum_{j<\ell} w(j) P_{j\ell} d^2(j,\ell) + \sum_{j \in [s]} w(j) d_j^+ \sum_{\ell > j} P_{j\ell} d^2(j,\ell) - \sum_{j \in [s]} w(j) d_j^+ d^2(j,i) \\
&\quad = \hat{f} + 2\Phi + \sum_{j<\ell} w(j) P_{j\ell} \|x_j - x_\ell\|^2 + \sum_{j \in [s]} w(j) d_j^+ \sum_{\ell > j} P_{j\ell} \|x_j - x_\ell\|^2 - \sum_{j \in [s]} w(j) d_j^+ \|x_j\|^2
\end{aligned}$$

Because of Corollary 5.9, the first term after $\hat{f} + 2\Phi$ can be upper bounded by Φ. Next, we will focus on upper bounding the remaining term (shown below) by $\ln(2) \cdot \Phi$.

$$\sum_{j \in [s]} w(j) d_j^+ \cdot \sum_{\ell > j} P_{j\ell} \cdot \|x_j - x_\ell\|^2 - \sum_{j \in [s]} w(j) d_j^+ \cdot \|x_j\|^2 \ .$$

Using Lemma 5.11, for any $\theta \in [0,1]$, we have

$$\begin{aligned}
&\sum_{j<\ell} w(j) d_j^+ P_{j\ell} \|x_j - x_\ell\|^2 - \sum_{j\in[s]} w(j) d_j^+ \|x_j\|^2 \\
&\quad \le \theta \sum_{j<\ell} w(j) P_{j\ell} \cdot \|x_j - x_\ell\|^2 + \sum_{j<\ell} [d_j^+ - \theta]^+ \cdot w(j) P_{j\ell} \|x_j - x_\ell\|^2 - \sum_{j\in[s]} w(j) d_j^+ \|x_j\|^2 \\
&\quad \le \theta \cdot \Phi + \underbrace{\sum_{j<\ell} [d_i^+ - \theta]^+ \cdot w(j) P_{j\ell} \|x_j - x_\ell\|^2 - \sum_{j\in[s]} w(j) d_j^+ \|x_j\|^2}_{S_2} \qquad \text{(Lemma 5.8)}
\end{aligned}$$

For the term S_2, we derive the following upper bound:

$$\begin{aligned}
S_2 &= \sum_{j\in[s]:d_j^+>\theta} \sum_{\ell>j} (d_j^+ - \theta) w(j) P_{j\ell} \cdot \|x_j - x_\ell\|^2 - \sum_{j\in[s]} w(j) d_j^+ \|x_j\|^2 \\
&\le \sum_{j\in[s]:d_j^+>\theta} \sum_{\ell>j} (d_j^+ - \theta) w(j) P_{j\ell} \cdot \left(\frac{\|x_j\|^2}{d_j^+ - \theta} + \frac{\|x_\ell\|^2}{1 - d_j^+ + \theta} \right) - \sum_{j\in[s]} w(j) d_j^+ \|x_j\|^2 \\
&\qquad \text{(Lemma 3.4)} \\
&\le \sum_{j\in[s]:d_j^+>\theta} w(j) \left(\sum_{\ell>j} P_{j\ell} \right) \|x_j\|^2 + \sum_{j\in[s]:d_j^+>\theta} \sum_{\ell>j} \frac{d_j^+ - \theta}{1 - d_j^+ + \theta} \cdot w(j) P_{j\ell} \|x_\ell\|^2 - \sum_{j\in[s]} w(j) d_j^+ \|x_j\|^2 \\
&\le \sum_{j\in[s]:d_j^+>\theta} \sum_{\ell>j} \frac{d_j^+ - \theta}{1 - d_j^+ + \theta} \cdot w(j) P_{j\ell} \|x_\ell\|^2 \qquad \text{(definition of } d_j^+\text{)} \\
&= \sum_{\ell\in[s]} \|x_\ell\|^2 \cdot \sum_{j<\ell:d_j^+>\theta} \frac{d_j^+ - \theta}{1 - d_j^+ + \theta} \cdot w(j) P_{j\ell} \ .
\end{aligned}$$

Let $g(y) = \frac{y-\theta}{1-y+\theta}$ for any $y \in [\theta, 1]$. This is easy to verify that $g(\theta) = 0$. Therefore, we have $g(y) = \int_\theta^y g'(z) dz$. Hence, we can further upper bound S_2 by

$$\begin{aligned}
S_2 &\le \sum_{\ell\in[s]} \|x_\ell\|^2 \cdot \sum_{j<\ell:d_j^+>\theta} \int_\theta^{d_j^+} g'(z) \cdot w(j) P_{j\ell} dz \\
&= \sum_{\ell\in[s]} \|x_\ell\|^2 \cdot \int_{z=\theta}^1 g'(z) \cdot \sum_{j<\ell:d_j^+>z} w(j) P_{j\ell} dz \ . \\
&\le \sum_{\ell\in[s]} \|x_\ell\|^2 \cdot \int_{z=\theta}^1 g'(z) \cdot w(\ell)(1-z) dz \qquad \text{(Lemma 5.11)} \\
&= \sum_{\ell\in[s]} w(\ell) \|x_\ell\|^2 \cdot \int_{z=\theta}^1 g'(z)(1-z) dz
\end{aligned}$$

Since $g(z) = 0$ when $z = \theta$ and $1 - z = 0$ when $z = 1$, we have

$$\int_{z=\theta}^1 g'(z) \cdot (1-z) dz = \int_{z=\theta}^1 g(z) dz = \int_{z=\theta}^1 \frac{z-\theta}{1-z+\theta} dz = \theta - 1 - \ln(\theta) \ .$$

We have $S_2 \le (\theta - 1 - \ln(\theta)) \cdot \Phi$. Therefore, we have

$$\sum_{j<\ell} w(j) d_j^+ P_{j\ell} \|x_j - x_\ell\|^2 - \sum_{j\in[s]} w(j) d_j^+ \|x_j\|^2 \le (\theta + \theta - 1 - \ln(\theta)) \cdot \Phi$$
$$= (2\theta - 1 - \ln(\theta)) \cdot \Phi \ .$$

If we choose $\theta = 1/2$, we get our desired bound that

$$\sum_{j<\ell} w(j) d_j^+ P_{j\ell} \|x_j - x_\ell\|^2 - \sum_{j\in[s]} w(j) d_j^+ \|x_j\|^2 \le \ln(2) \cdot \Phi \ .$$

5.3.5 Step 4(b): Proof for the general metric case

In this part, we will prove Lemma 4.11 for the general metric case. We will consider $\Gamma = 5$ here. For simplicity, we use $a_j = d^2(j,i)$ for each $j \in [s]$ in this part. According to Equation (10) and Lemma 5.7, we have

$$\sum_{j\in[s]} w(j) \left(\alpha_j^* - (\rho-1)R_j^2\right)$$
$$\le \hat{f} + 2\Phi + \sum_{j<\ell} w(j) P_{j\ell} d^2(j,\ell) + \sum_{j\in[s]} w(j) d_j^+ \sum_{\ell>j} P_{j\ell} d^2(j,\ell) - \sum_{j\in[s]} w(j) d_j^+ d^2(j,i)$$

First, because of Corollary 5.9, we can upper bound the first term after $\hat{f} + 2\Phi$ by 2Φ. Next, we upper bound the remaining term, i.e.,

$$\sum_{j\in[s]} w(j) d_j^+ \sum_{\ell>j} P_{j\ell} d^2(j,\ell) - \sum_{j\in[s]} w(j) d_j^+ d^2(j,i) \tag{19}$$

by Φ for the general metric case.

Using the definition of a_j, we have

$$\begin{aligned}
\textit{Equation (19)} &\le \sum_{j\in[s]} w(j) d_j^+ \sum_{\ell>j} P_{j\ell} \cdot 2(a_j + a_\ell) - \sum_{j\in[s]} w(j) d_j^+ a_j \\
&\qquad\qquad \text{(definition of } a_j \text{ and Lemma 3.4)} \\
&= 2\sum_{j<\ell} w(j) d_j^+ P_{j\ell} \cdot a_\ell + 2\sum_{j<\ell} w(j) d_j^+ P_{j\ell} \cdot a_j - \sum_{j\in[s]} w(j) d_j^+ a_j \\
&= 2\sum_{j<\ell} w(j) d_j^+ P_{j\ell} \cdot a_\ell + \sum_{j\in[s]} w(j)(2(d_j^+)^2 - d_j^+) a_j \qquad \text{(definition of } d_j^+) \\
&= 2\sum_{j<\ell} w(j) \left(\int_{\theta=0}^{1} \mathbb{1}(d_j^+ > \theta) d\theta \right) \cdot P_{j\ell} a_\ell + \sum_{j\in[s]} w(j)(2(d_j^+)^2 - d_j^+) a_j \\
&= 2\sum_{\ell\in[s]} \left(\int_{\theta=0}^{1} \sum_{j<\ell} \mathbb{1}(d_j^+ > \theta) w(j) P_{j\ell} d\theta \right) \cdot a_\ell + \sum_{j\in[s]} w(j)(2(d_j^+)^2 - d_j^+) a_j
\end{aligned}$$

Note that for any $\ell \in [s]$, the coefficient of a_ℓ in the first term can be upper bounded as follows:

$$\begin{aligned}\int_{\theta=0}^{1} \sum_{j<\ell} \mathbb{1}(d_j^+ > \theta) w(j) P_{j\ell} d\theta &\le \int_{\theta=0}^{d_\ell^+} \sum_{j<\ell} w(j) P_{j\ell} d\theta + \int_{\theta=d_\ell^+}^{1} \sum_{j<\ell} \mathbb{1}(d_j^+ > \theta) w(j) P_{j\ell} d\theta \\ &\le d_\ell^+ \cdot w(\ell)(1-d_\ell^+) + \int_{\theta=d_\ell^+}^{1} w(\ell)(1-\theta) d\theta \\ &\qquad \text{(Corollary 5.10 and Lemma 5.11)} \\ &= w(\ell)\left(d_\ell^+(1-d_\ell^+) + \frac{(1-d_\ell^+)^2}{2}\right)\end{aligned}$$

Therefore, we can further upper bound Equation (19) by (the final inequality is because $d_j^+ \in [0,1]$ for each $j \in [s]$):

$$\begin{aligned}\textit{Equation (19)} &\le \sum_{j\in[s]} w(j) \cdot \left(2d_j^+(1-d_j^+) + (1-d_j^+)^2 + 2(d_j^+)^2 - d_j^+\right) \cdot a_j \\ &= \sum_{j\in[s]} w(j) \cdot \left(1 - d_j^+ + (d_j^+)^2\right) \cdot a_j \le \Phi\ .\end{aligned}$$

5.4 A generalization of Lemma 4.11 for the Euclidean case

In this subsection, we will prove a generalization of Lemma 4.11, where we only have some approximate version of the constraints. We will show that the same guarantee can be achieved up to a small additive factor. Later, we will use this generalization to obtain the bicriteria approximation guarantee for the Euclidean case. Because the proof is highly technical with dealing with the small additive factor and mostly follows the spirit of the earlier derivations, we defer it to Subsection B.2.

Lemma 5.20. *Let $\rho = 2$ and $\Gamma = 3 + \ln(2)$. Let $\hat{f}$ be any non-negative real number. Let $\varepsilon \ge 0$ be any sufficiently small non-negative real number. Let $([s] \cup \{i\}, d)$ be a Euclidean metric. Let $w : [s] \to \mathbb{R}_{>0}$ be any non-negative weight function. Let $a \in [s]$ be any positive integer such that $a \le s$. Let $(IC^t)_{t\in[s]}$ and $(DC^t)_{t\in[s]}$ be two families of sets satisfying:*

(a) For each $t \in [s]$, $DC^t \cap IC^t = \emptyset$ and $DC^t \cup IC^t = [t-1]$.

(b) For each $t \in [s-1]$, $DC^t \subseteq DC^{t+1}$.

(c) $[a+1 : s-1] \subseteq IC^s$.

Suppose that the variables $(\alpha_j^*)_{j\in[s]}, (R_j)_{j\in[s]}, (r_{jt})_{t\in[s], j\in DC^t}$ satisfy the following constraints

(1) For any $t \in [s], j \in DC^t$, we have $r_{jt} \le R_j$.

(2) For each $t \in [a], j \in DC^t$ and $\ell \in IC^t \cup \{t\}$, we have $\alpha_\ell^* \le (r_{jt} + d(j,\ell))^2$.

(3) For each $t \in [a] \cup \{s\}$, we have

$$\sum_{\ell\in[t:s]} w(\ell)\alpha_t^* + \sum_{\ell\in IC^t} w(\ell)\alpha_\ell^* + \sum_{j\in DC^t} w(j) r_{jt}^2 \le \hat{f} + \sum_{\ell\in IC^t\cup[t:s]} \rho w(\ell) d^2(\ell,i) + \sum_{j\in DC^t} w(j) d^2(j,i).$$

(4) There is a set $L \subseteq [a] \cap IC^s$ such that:

- $\sum_{\ell\in L} w(\ell) \le \varepsilon \cdot \sum_{\ell\in[a+1:s]} w(\ell)$;

- *for any $j \in DC^s, \ell \in (IC^s \cup \{s\}) \setminus L$, we have $\alpha^*_\ell \le (1+\varepsilon)\cdot(r_{js} + d(j,\ell))^2$;*
- *for any $j \in DC^s, \ell \in L$ and $\ell' \in [a+1:s]$, we have $\alpha^*_\ell \le (r_{js} + d(j,\ell'))^2$.*

Then, the following holds:

$$\sum_{j\in[s]} w(j)\cdot(\alpha^*_j - (\rho-1)R_j^2) \le \hat{f} + (\Gamma + 10\varepsilon)\cdot\sum_{j\in[s]} w(j)\cdot d^2(j,i)\ . \tag{20}$$

Remark 5.21. *This is a generalization of Lemma 4.11, because Lemma 4.11 is a special case when $\varepsilon = 0, a = s-1$ and $L = \emptyset$.*

6 An Improved LMP 4.9-Approximation for the Metric Case

In this section, we will prove Lemma 4.12, restated below for convenience.

AS: i dont think we really need to restate the Lemma 4.12 here again, it's right before the section starts

Lemma 4.12. *Let $\rho = 2.5$ and $\Gamma = 4.9$. Let $\hat{f}$ be any non-negative real number. Let $([s]\cup\{i\}, d)$ be a metric. Let $w : [s] \to \mathbb{R}_{>0}$ be any positive weight function. Let $(IC^t)_{t\in[s]}$ and $(DC^t)_{t\in[s]}$ be two families of sets satisfying:*

(a) For each $t \in [s]$, $DC^t \cap IC^t = \emptyset$ and $DC^t \cup IC^t = [t-1]$.

(b) For each $t \in [s-1]$, $DC^t \subseteq DC^{t+1}$.

Suppose that the variables $(\alpha^*_j)_{j\in[s]}, (R_j)_{j\in[s]}, (r_{jt})_{t\in[s], j\in DC^t}$ satisfy the following constraints

(1) For any $t \in [s], j \in DC^t$, we have $r_{jt} \le R_j$.

(2) For each $t \in [s], j \in DC^t$, we have $\alpha^*_t \le (r_{jt} + d(j,t))^2$.

(3) For each $t \in [s]$, we have

$$\sum_{\ell\in[t:s]} w(\ell)\alpha^*_t + \sum_{\ell\in IC^t} w(\ell)\alpha^*_\ell + \sum_{j\in DC^t} w(j) r_{jt}^2 \le \hat{f} + \sum_{\ell\in IC^t\cup[t:s]} \rho w(\ell) d^2(\ell,i) + \sum_{j\in DC^t} w(j) d^2(j,i).$$

Then, the following holds:

$$\sum_{j\in[s]} w(j)\cdot(\alpha^*_j - (\rho-1)R_j^2) \le \hat{f} + \Gamma\cdot\sum_{j\in[s]} w(j)\cdot d^2(j,i)\ . \tag{9}$$

[AS:Here the flow is for now unweighted full proof, then stating the weighted proof, with the differences in the bookkeeping required highlighted. If we plan to keep just the weighted part, we can input the metric-lmp-weighted-full.tex commented out below.]

[AS:adding comments from Moses from the weighted-full.tex which is currently commented out: (1) the notation of suffix sums that used Σ_t was confusing with the $\sum$. So changed it to W_t. (2) Adding an overview of the proof soon. (3) no forward referencing the equations.]

MC: Note that metric-lmp-unweighted and metric-lmp-weighted are not commented out currently[AS:fixed]

6.1 Unweighted

AS: defined the macros for a, η, β here, can change later.

In this subsection, we prove Lemma 4.12 in the special case $w(\ell) = 1$ for all $\ell \in [s]$. Define $a_t \stackrel{\text{def}}{=} d(t,i)$ for all $t \in [s]$. The final proof of this lemma for the unweighted case was obtained by GPT 5.2/5.4 Pro/Thinking, after several rounds of exchange with the authors implementing various theoretical ideas and experiments, and revised by the authors.

As briefly discussed in the overview in Subsection 2.1, the proof has four steps. First, from the inequalities at time $\alpha_t - \varepsilon$ given in condition (3) of Lemma 4.12, we derive a family of constraints (P_t) by using the triangle inequality to lower bound the terms involving r_{jt}. Second, we choose coefficients $(\beta)_{t\in[s]}$ so that, in the weighted combination of these constraints, the coefficient of each α_t^* is exactly 1. Third, after collecting terms, we obtain a quadratic form in the distances to the facility i. Finally, we bound the top eigenvalue of the corresponding matrix by the Collatz–Wielandt Formula, which yields the claimed constant.

MC: The constraint (I_t) has a natural interpretation: the total payments towards opening a facility never exceed $\hat{f}$. Referring to it as such could help the reader. Could also label this inequality with something like: facility payment upper bound, then use this label to refer to the inequality.

AS: i agree, but then it should be condition(3), not (I_t) imo
MC: yes, good point!

Step 1: From (I_t) to (P_t). For each $t \in [s]$, given $w(t) = 1$ and $a_t \stackrel{\text{def}}{=} d(t,i)$ for all $t \in [s]$, rewriting and rearranging the condition (3) gives

$$\sum_{j\in[t:s]} \alpha_t^* + \sum_{j\in IC^t} \alpha_j^* + \sum_{j\in DC^t} w(j) r_{jt}^2 \le \hat{f} + \sum_{j\in IC^t\cup[t:s]} \rho d^2(j,i) + \sum_{j\in DC^t} w(j) d^2(j,i).$$

(Condition (3))

$$\implies \sum_{j\in DC^t} (r_{jt}^2 - a_j^2) + \sum_{j\in IC^t} (\alpha_j^* - \rho a_j^2) + \sum_{j\ge t} (\alpha_t^* - \rho a_j^2) \le \hat{f}. \tag{I_t}$$

Fix $t \in [s]$. For each $j \in DC^t$, conditions (1)–(2) and the triangle inequality imply

$$\alpha_t^* \le (r_{jt} + d(j,t))^2 \le (r_{jt} + a_j + a_t)^2 \le r_{jt}^2 + (a_j + a_t)^2 + 2(a_j + a_t) R_j.$$

Rearranging and using $2a_j a_t \le a_j^2 + a_t^2$, we obtain

$$r_{jt}^2 - a_j^2 \ge \alpha_t^* - 2a_t^2 - 3a_j^2 - 2(a_j + a_t) R_j.$$

Substituting into Equation (I_t) gets us

$$\sum_{j\in DC^t} \alpha_t^* + \sum_{j\in IC^t} \alpha_j^* + \sum_{j\ge t} \alpha_t^* - \hat{f} \le \sum_{j\in DC^t} \left(2a_t^2 + 3a_j^2 + 2(a_t + a_j) R_j\right) + \sum_{j\in IC^t} \rho a_j^2 + \sum_{j\ge t} \rho a_j^2. \tag{P_t}$$

Step 2: Choosing coefficients. We now choose nonnegative coefficients $\beta_1, \ldots, \beta_s$ and multiply each inequality (P_t) with β_t and sum over all $t \in [s]$ such that the coefficient of each α_t^* is exactly 1. This is very similar to the proof of how coefficients are derived in [CCG+25].

AS: i think it is better to push this proof of obtaining the weights to appendix, since it is the same as [CCG+25]

Lemma 6.1. *For each $t \in [s]$, define*

$$m_t \stackrel{def}{=} |DC^t|, \qquad c_t \stackrel{def}{=} s - |IC^t| = s - t + 1 + m_t, \qquad g_t \stackrel{def}{=} \sum_{u\ge t} \beta_u,$$

$$\eta_t \stackrel{def}{=} \sum_{u>t:\, t\in DC^u} \beta_u, \qquad A_t \stackrel{def}{=} m_t \beta_t.$$

Then there exist nonnegative coefficients $\beta_1, \ldots, \beta_s$ such that, for every $t \in [s]$,

$$\beta_t c_t + \sum_{u>t:\, t\in IC^u} \beta_u = 1, \tag{21}$$

and moreover,

$$0 \le \eta_t \le g_t, \qquad \sum_{t=1}^{s} \beta_t = 1, \qquad A_t + g_t \le 1.$$

MC: I agree

AS: is it unclear on what "weighted combination" means here?

MC: Yes, clarify that β_t is the multiplier of (P_t)[AS:done]

AS: we had also defined t_t as the first time t becomes DC, i dont think that's needed either

Proof. Define the coefficients recursively, for $t = s, s-1, \ldots, 1$, by

$$\beta_t = \frac{1 - \sum_{u>t:\, t\in IC^u} \beta_u}{c_t}. \tag{22}$$

Since Equation (21) is triangular, this is the unique solution of the system. We verify that the resulting coefficients are nonnegative.

Fix $t \in \{0\} \cup [s]$ and consider the suffix $\{\alpha^*_{t+1}, \ldots, \alpha^*_s\}$. In the final weighted combination, its total coefficient is $s - t$. On the other hand, for each $u > t$, the contribution of the u'th inequality Equation (P_t) to this suffix is

$$c_u + |IC^u \cap [t+1, u-1]|.$$

Since IC^u and DC^u partition $[u-1]$, we have

$$\begin{aligned} c_u + |IC^u \cap [t+1, u-1]| &= (s-u+1) + m_u + (u-t-1) - |DC^u \cap [t+1, u-1]| \\ &= s - t + |DC^u \cap [1, t]| \\ &\geq s - t + m_t, \end{aligned}$$

where we used $DC^t \subseteq DC^u$ and $DC^t \subseteq [t-1]$. Summing over $u > t$ gives

$$(s - t + m_t) \sum_{u>t} \beta_u \leq s - t, \tag{23}$$

and hence

$$\sum_{u>t} \beta_u \leq \frac{s-t}{s-t+m_t}.$$

Therefore,

$$\sum_{u>t:\, t\in IC^u} \beta_u \leq \sum_{u>t} \beta_u \leq \frac{s-t}{s-t+m_t} < 1,$$

so Equation (22) indeed yields $\beta_t \geq 0$.

The inequality $0 \leq \eta_t \leq g_t$ is immediate from the definitions.

To prove $\sum_{t=1}^s \beta_t = 1$, sum the coefficients of all α^*-variables in the weighted combination of Equation (P_t). Each inequality contributes total coefficient s, while by construction each α^*_u has coefficient 1. Hence

$$s \sum_{t=1}^{s} \beta_t = s,$$

so $\sum_{t=1}^s \beta_t = 1$, and in particular $g_t \leq 1$.

Finally, fix $t \in [s]$ and consider the suffix $\{\alpha^*_t, \ldots, \alpha^*_s\}$. Its total coefficient in the final weighted combination is $s - t + 1$. For each $u \geq t$, the contribution of the u'th inequality to this suffix is

$$c_u + |IC^u \cap [t, u-1]|.$$

Again using that IC^u and DC^u partition $[u-1]$, we obtain

$$\begin{aligned} c_u + |IC^u \cap [t, u-1]| &= (s-u+1) + m_u + (u-t) - |DC^u \cap [t, u-1]| \\ &= s - t + 1 + |DC^u \cap [1, t-1]| \\ &\geq s - t + 1 + m_t = c_t. \end{aligned}$$

Summing over $u \geq t$ yields $c_t g_t \leq s - t + 1$, so

$$g_t \leq \frac{s - t + 1}{c_t}. \tag{24}$$

Since Equation (21) implies $c_t \beta_t \leq 1$, we also have $\beta_t \leq 1/c_t$. Therefore,

$$A_t + g_t = m_t \beta_t + g_t \leq \frac{m_t}{c_t} + \frac{s - t + 1}{c_t} = 1.$$

□

Step 3: The quadratic form We next compute the quadratic form produced by the weighted sum of $(P_t)_{t \in [s]}$.

Lemma 6.2. *For each $t \in [s]$ and each $x \in \mathbb{R}^s$, define $U_t(x) \stackrel{def}{=} \sum_{u:\, t \in DC^u} \beta_u x_u$. Then the weighted sum of all inequalities Equation (P_t) over $t \in [s]$ implies*

$$\sum_{t \in [s]} \alpha_t^* - (\rho - 1) R_t^2 - \hat{f} \leq \rho \sum_{t \in [s]} a_t^2 + Q_\rho(a),$$

where, for every $x \in \mathbb{R}^s$,

$$Q_\rho(x) = \sum_{t \in [s]} \big(2A_t + (3 - \rho)\eta_t\big) x_t^2 + \frac{1}{\rho - 1} \sum_{t \in [s]} \big(U_t(x) + \eta_t x_t\big)^2. \tag{25}$$

AS: we can replace $\mathbb{R}$ with $\mathbb{R}_{\geq 0}$ here, if it's confusing. but the statement is true in general as well.

Proof. By Equation (21), the weighted sum of the left-hand sides of $(P_t)_{t \in [s]}$ is $\sum_{t \in [s]} \alpha_t^* - \hat{f}$. Let $\mathcal{R}$ denote the weighted sum of the right-hand sides. Then

$$\sum_{t \in [s]} \alpha_t^* - \hat{f} \leq \mathcal{R}.$$

We now collect the terms in $\mathcal{R}$.

First,

$$\sum_{t \in [s]} \beta_t \sum_{j \in DC^t} 2a_t^2 = \sum_{t \in [s]} 2A_t a_t^2.$$

Second,

$$\sum_{t \in [s]} \beta_t \sum_{j \in DC^t} 3a_j^2 = \sum_{t \in [s]} 3\eta_t a_t^2.$$

Third,

$$\sum_{t \in [s]} \beta_t \sum_{j \in DC^t} 2(a_t + a_j) R_j = 2 \sum_{t \in [s]} R_t \big(U_t(a) + \eta_t a_t\big).$$

Finally, for each fixed t, the sets

$$\{u \in [s] : t \in IC^u\}, \qquad \{u \in [s] : t \in DC^u\}, \qquad \{u \in [s] : u \leq t\}$$

partition $[s]$. Since $\sum_{t=1}^s \beta_t = 1$ by Lemma 6.1, it follows that

$$\sum_{u:\, t \in IC^u} \beta_u + \sum_{u \leq t} \beta_u = 1 - \eta_t,$$

AS: initially i did all this in a series of inequalities within align environment, that was a lil concise, but since Aditya pointed out we need to make it less scary, i divided it into parts.

and hence

$$\sum_{u\in[s]}\beta_u\left(\sum_{j\in IC^u}\rho a_j^2+\sum_{j\geq u}\rho a_j^2\right)=\rho\sum_{t\in[s]}(1-\eta_t)a_t^2.$$

Therefore,

$$\mathcal{R}=\sum_{t\in[s]}2A_ta_t^2+\sum_{t\in[s]}3\eta_ta_t^2+2\sum_{t\in[s]}R_t\big(U_t(a)+\eta_ta_t\big)+\rho\sum_{t\in[s]}(1-\eta_t)a_t^2.$$

Equivalently,

$$\mathcal{R}=\rho\sum_{t\in[s]}a_t^2+\sum_{t\in[s]}\big(2A_t+(3-\rho)\eta_t\big)a_t^2+2\sum_{t\in[s]}R_t\big(U_t(a)+\eta_ta_t\big).$$

Hence

$$\sum_{t\in[s]}\alpha_t^*-(\rho-1)R_t^2-\hat{f}\leq\rho\sum_{t\in[s]}a_t^2+\sum_{t\in[s]}\big(2A_t+(3-\rho)\eta_t\big)a_t^2+\sum_{t\in[s]}\Big(2R_t\big(U_t(a)+\eta_ta_t\big)-(\rho-1)R_t^2\Big).$$

As in the warm-up, we get rid of the mixed term R using

$$2Rx-(\rho-1)R^2\leq\frac{x^2}{\rho-1}\qquad(R,x\in\mathbb{R}).$$

Applying this with $x=U_t(a)+\eta_ta_t$ for each t, we obtain

$$\sum_{t\in[s]}\alpha_t^*-(\rho-1)R_t^2-\hat{f}\leq\rho\sum_{t\in[s]}a_t^2+\sum_{t\in[s]}\big(2A_t+(3-\rho)\eta_t\big)a_t^2+\frac{1}{\rho-1}\sum_{t\in[s]}\big(U_t(a)+\eta_ta_t\big)^2.$$

This is exactly the claimed inequality. □

Step 4: Matrix Form It remains to bound Q_ρ.

Lemma 6.3. *Let K be the unique symmetric matrix such that $Q_\rho(x)=x^\top Kx$ for all $x\in\mathbb{R}^s$. Then, for every $t\in[s]$ and every $x\in\mathbb{R}^s_{\geq 0}$,*

$$(Kx)_t=\Big(2A_t+(3-\rho)\eta_t+\frac{\eta_t^2}{\rho-1}\Big)x_t+\frac{\eta_t}{\rho-1}U_t(x)+\frac{\beta_t}{\rho-1}\sum_{u\in DC^t}\big(U_u(x)+\eta_ux_u\big).$$

Moreover, K is entrywise nonnegative for all $1<\rho\leq 3$. Consequently,

$$Q_\rho(x)\leq\lambda_{\max}(K)\|x\|_2^2\qquad\text{for all } x\in\mathbb{R}^s_{\geq 0}.$$

Proof. Since $Q_\rho(x)=x^\top Kx$ and K is symmetric, we have

$$\frac{\partial Q_\rho(x)}{\partial x_t}=2(Kx)_t.$$

Differentiating Equation (25), we obtain

$$\begin{aligned}(Kx)_t&=\big(2A_t+(3-\rho)\eta_t\big)x_t+\frac{\eta_t}{\rho-1}\big(U_t(x)+\eta_tx_t\big)+\frac{\beta_t}{\rho-1}\sum_{u\in DC^t}\big(U_u(x)+\eta_ux_u\big)\\&=\Big(2A_t+(3-\rho)\eta_t+\frac{\eta_t^2}{\rho-1}\Big)x_t+\frac{\eta_t}{\rho-1}U_t(x)+\frac{\beta_t}{\rho-1}\sum_{u\in DC^t}\big(U_u(x)+\eta_ux_u\big).\end{aligned}$$

Every coefficient in this expression is nonnegative when $1<\rho\leq 3$, so K is entrywise nonnegative. The final claim follows from the definition $\lambda_{\max}(K)$. □

We now pick $\rho = 5/2$ and replace η_t by g_t, since $\eta_t \le g_t$ (Lemma 6.1). For every $t \in [s]$ and every $x \in \mathbb{R}^s_{\ge 0}$, Lemma 6.3 gives

$$(Kx)_t \le \Big(2A_t + \frac{1}{2}g_t + \frac{2}{3}g_t^2\Big)x_t + \frac{2}{3}g_t U_t(x) + \frac{2}{3}\beta_t \sum_{u \in DC^t} U_u(x) + \frac{2}{3}\beta_t \sum_{u \in DC^t} g_u x_u. \tag{26}$$

The next lemma applies the Collatz–Wielandt Formula [Mey00] with a specific test vector.

AS: citing a book here, the original work is in german

Lemma 6.4. *Fix $\tau = \sqrt[3]{12}$. Then*

$$\lambda_{\max}(K) \le \max_{1 \le x \le \tau} F(x),$$

where

$$F(x) = \frac{48x^6 - 47\tau x^5 - 357x^3 + (564\tau - 396)x^2 + 4665}{2178}. \tag{27}$$

Proof. We choose $h \in \mathbb{R}^s_{\ge 0}$ by

$$h_t = (1 + 11g_t)^{-2/3} \qquad (t \in [s]).$$

By the min–max Collatz–Wielandt Formula [Mey00, Theorem 8.3.3] for non-negative symmetric matrix $K \ge 0$, with spectral radius $\rho(K)$, we have

$$\lambda_{\max}(K) = \rho(K) = \min_{x \in \mathbb{R}^s_{\ge 0}} \max_{t \in [s], x_t \ne 0} \frac{(Kx)_t}{x_t} \le \max_{t \in [s]} \frac{(Kh)_t}{h_t} \qquad \text{(Collatz–Wielandt Formula)}$$

Claim 6.5. *For every $t \in [s]$,*

$$\frac{2g_t U_t(h)}{3h_t} \le \frac{2}{11} g_t \Big((1 + 11g_t) - (1 + 11g_t)^{2/3}\Big), \tag{28}$$

$$\frac{2\beta_t}{3h_t} \sum_{u:\, t \in DC^u} g_u h_u \le A_t \cdot \frac{2}{3}(1 + 11g_t)^{2/3}\tau^{-2}, \tag{29}$$

$$\frac{2\beta_t}{3h_t} \sum_{u:\, t \in DC^u} U_u(h) \le A_t (1 + 11g_t)^{2/3} \cdot \frac{2}{11}(\tau - 1). \tag{30}$$

Consequently,

$$\frac{(Kh)_t}{h_t} \le G(g_t), \tag{31}$$

where

$$\begin{aligned} G(g) \stackrel{def}{=}\ & 2(1-g) + \frac{1}{2}g + \frac{2}{3}g^2 + \frac{2}{3}(1-g)(1+11g)^{2/3}\tau^{-2} \\ & + \frac{2}{11}(1-g)(1+11g)^{2/3}(\tau - 1) + \frac{2}{11}g\Big((1+11g) - (1+11g)^{2/3}\Big). \end{aligned} \tag{32}$$

Proof. Since the sets DC^u are monotone in u, the set $\{u \in [s] : t \in DC^u\}$ is a suffix of $\{t+1, \dots, s\}$. Let u_t be its first element. Let $u_t = s + 1$ if $t \notin DC^u$ for every u. Then

$$U_t(h) = \sum_{u \ge u_t} \beta_u h_u = \sum_{u \ge u_t} (g_u - g_{u+1})(1 + 11g_u)^{-2/3}.$$

Since the function $z \mapsto (1+11z)^{-2/3}$ is decreasing, this sum is bounded by the corresponding integral:

$$U_t(h) \le \int_0^{\eta_t} (1+11z)^{-2/3}\,dz \le \frac{3}{11}\Big((1+11\eta_t)^{1/3}-1\Big). \tag{33}$$

Together with $\eta_t \le g_t$, this gives Equation (28).

For Equation (30), Equation (33) implies

$$U_u(h) \le \frac{3}{11}\Big((1+11\eta_u)^{1/3}-1\Big) \le \frac{3}{11}(\tau-1),$$

since $\eta_u \in [0,1]$. Therefore,

$$\frac{2\beta_t}{3h_t}\sum_{u:\,t\in DC^u} U_u(h) \le \frac{2\beta_t}{3}(1+11g_t)^{2/3}\cdot m_t \cdot \frac{3}{11}(\tau-1) = A_t(1+11g_t)^{2/3}\cdot\frac{2}{11}(\tau-1).$$

For Equation (29), the function $g \mapsto g(1+11g)^{-2/3}$ is increasing on $[0,1]$, so

$$g_u h_u = g_u(1+11g_u)^{-2/3} \le 12^{-2/3} = \tau^{-2}.$$

Hence

$$\sum_{u:\,t\in DC^u} g_u h_u \le m_t \tau^{-2},$$

and therefore

$$\frac{2\beta_t}{3h_t}\sum_{u:\,t\in DC^u} g_u h_u \le \frac{2\beta_t m_t}{3}(1+11g_t)^{2/3}\tau^{-2} = A_t\cdot\frac{2}{3}(1+11g_t)^{2/3}\tau^{-2}.$$

Combining Equations (26) and (28) to (30) and using $A_t + g_t \le 1$ from Lemma 6.1, we obtain

$$\begin{aligned}
\frac{(Kh)_t}{h_t} &\le \Big(2A_t + \frac{1}{2}g_t + \frac{2}{3}g_t^2\Big) + \frac{2g_tU_t(h)}{3h_t} + \frac{2\beta_t}{3h_t}\sum_{u:\,t\in DC^u} U_u(h) + \frac{2\beta_t}{3h_t}\sum_{u:\,t\in DC^u} g_u h_u \\
&\le 2A_t + \frac{1}{2}g_t + \frac{2}{3}g_t^2 + \frac{2}{11}g_t\Big((1+11g_t) - (1+11g_t)^{2/3}\Big) \\
&\quad + A_t(1+11g_t)^{2/3}\cdot\frac{2}{11}(\tau-1) + A_t\cdot\frac{2}{3}(1+11g_t)^{2/3}\tau^{-2} \\
&\le G(g_t).
\end{aligned}$$

□

Finally, we simplify the expression in Equation (31) by substituting $g = (x^3-1)/11$ and simplifying, using $\tau^{-2} = \tau/12$. This gives

$$\frac{(Kh)_t}{h_t} \le G(g_t) = F(x),$$

where

$$F(x) = \frac{48x^6 - 47\tau x^5 - 357x^3 + (564\tau - 396)x^2 + 4665}{2178}.$$

□

The remaining task is to prove an upper bound on $F(x)$ as follows:

Lemma 6.6. $\max_{x\in[1,\tau]} F(x) < \frac{12}{5}$.

Proof. This is deferred to Subsection C.1. □

We can now finish the proof of the unweighted version of Lemma 4.12. Using Lemma 6.2 with $\rho = 5/2$, we obtain

$$\begin{aligned}
\sum_{t\in[s]} \alpha_t^* - (\rho-1)R_t^2 - \hat{f} &\le \frac{5}{2}\sum_{t\in[s]} a_t^2 + Q_{5/2}(a) && \text{(Lemma 6.2)}\\
&\le \frac{5}{2}\sum_{t\in[s]} a_t^2 + \lambda_{\max}(K)\sum_{t\in[s]} a_t^2 && \text{(Lemma 6.3)}\\
&\le \frac{5}{2}\sum_{t\in[s]} a_t^2 + \left(\max_{x\in[1,\tau]} F(x)\right)\sum_{t\in[s]} a_t^2 && \text{(Lemma 6.4)}\\
&\le \frac{5}{2}\sum_{t\in[s]} a_t^2 + \frac{12}{5}\sum_{t\in[s]} a_t^2 && \text{(Lemma 6.6)}\\
&= 4.9\sum_{t\in[s]} a_t^2.
\end{aligned}$$

This proves Lemma 4.12 in the unweighted case, i.e., $w(\ell) = 1$ for all $\ell \in [s]$.

6.2 Weighted

AS: Since this subsection is really analogous to the unweighted part, I just state many proofs as just corollary of the above.

AS: again defined macros here, change as required to make things consistent between metric and euclidean

We now prove Lemma 4.12 in its full generality for any weight function $w : [s] \to \mathbb{R}_{>0}$.

To extend the argument to the weighted setting, the overall structure is the same as in the unweighted case: given the condition (3) of Lemma 4.12, we derive inequalities $(P_t^w)_{t\in[s]}$, choose coefficients so that the α_t^* terms sum up to $w(t)$ for each $t \in [s]$, rewrite the resulting bound as a quadratic form, and then upper bound its top eigenvalue. The new issue is the bookkeeping. The cardinalities of clients are replaced by weighted masses, the normalization of the coefficients changes accordingly. After introducing the appropriate weighted analogues of the unweighted parameters and rescaling by $\sqrt{w(t)}$, however, the same spectral argument goes through.

Since the proof is a weighted analogue of the unweighted subsection, we mention only the differences in the proof of the coefficients and in the quadratic form, and defer the full proof to Subsection C.2.

Define

$$a_t \stackrel{\text{def}}{=} d(t,i), \qquad W(S) \stackrel{\text{def}}{=} \sum_{j\in S} w(j), \qquad W_t \stackrel{\text{def}}{=} W([t:s]), \qquad M_t \stackrel{\text{def}}{=} W(DC^t).$$

For each $t \in [s]$, condition (3) gives

$$\sum_{j\in DC^t} w(j)(r_{jt}^2 - a_j^2) + \sum_{j\in IC^t} w(j)(\alpha_j^* - \rho a_j^2) + \sum_{j\ge t} w(j)(\alpha_t^* - \rho a_j^2) \le \hat{f}. \tag{I_t^w}$$

Exactly as in the unweighted case, for every $j \in DC^t$,

$$r_{jt}^2 - a_j^2 \ge \alpha_t^* - 2a_t^2 - 3a_j^2 - 2(a_j + a_t)R_j.$$

Substituting into Equation (I_t^w) yields

$$M_t\alpha_t^* + \sum_{j\in IC^t} w(j)\alpha_j^* + W_t\alpha_t^* - \hat{f}$$
$$\le 2M_t a_t^2 + \sum_{j\in DC^t} w(j)\big(3a_j^2 + 2(a_t+a_j)R_j\big) + \sum_{j\in IC^t}\rho w(j)a_j^2 + \sum_{j\ge t}\rho w(j)a_j^2. \qquad (P_t^w)$$

Lemma 6.7. *Define*

$$c_t \overset{def}{=} \frac{W_t + M_t}{w(t)}, \qquad g_t \overset{def}{=} \sum_{u\ge t}\beta_u, \qquad \eta_t \overset{def}{=} \sum_{u>t:\, t\in DC^u}\beta_u, \qquad A_t \overset{def}{=} \frac{M_t\beta_t}{w(t)}.$$

AS: is an analogue of these weighted coefficients defined in the stable part of [CCG+25]?

Then there exist nonnegative coefficients $\beta_1,\dots,\beta_s$ such that, for every $t\in[s]$,

$$\beta_t c_t + \sum_{u>t:\, t\in IC^u}\beta_u = 1, \qquad (34)$$

and moreover,

$$0 \le \eta_t \le g_t, \qquad \sum_{t=1}^{s}\beta_t = 1, \qquad A_t + g_t \le 1.$$

The proof is the same as that of Lemma 6.1, with cardinalities replaced by weighted masses. The only changes are the following two weighted analogues of Inequalities 23, 24

$$(W_{t+1}+M_t)\sum_{u>t}\beta_u \le W_{t+1}, \qquad (W_t+M_t)g_t \le W_t.$$

Lemma 6.8. *For $x\in\mathbb{R}^s$, define $U_t(x) \overset{def}{=} \sum_{u:\, t\in DC^u}\beta_u x_u$. Then*

$$\sum_{t\in[s]} w(t)\big(\alpha_t^* - (\rho-1)R_t^2\big) - \hat{f} \le \rho\sum_{t\in[s]} w(t)a_t^2 + Q_\rho^w(a),$$

where

$$Q_\rho^w(x) = \sum_{t\in[s]} w(t)\big(2A_t + (3-\rho)\eta_t\big)x_t^2 + \frac{1}{\rho-1}\sum_{t\in[s]} w(t)\big(U_t(x) + \eta_t x_t\big)^2. \qquad (35)$$

Proof. With the coefficients chosen as above, the same counting as in the unweighted proof gives

$$\sum_{t\in[s]} w(t)\alpha_t^* - \hat{f} \le \mathcal{R},$$

where

$$\mathcal{R} = \rho\sum_{t\in[s]} w(t)a_t^2 + \sum_{t\in[s]} w(t)\big(2A_t + (3-\rho)\eta_t\big)a_t^2 + 2\sum_{t\in[s]} w(t)R_t\big(U_t(a) + \eta_t a_t\big).$$

Subtracting $(\rho-1)\sum_t w(t)R_t^2$ and applying $2Rx - (\rho-1)R^2 \le x^2/(\rho-1)$ pointwise yields the claim. □

Lemma 6.9. *Let*

$$\widetilde{x} \stackrel{def}{=} (\sqrt{w(1)}x_1, \ldots, \sqrt{w(s)}x_s),$$

AS: essentially this is the only change (other than coefficients).

and let B be the symmetric matrix such that

$$Q^w_\rho(x) = \widetilde{x}^\top B \widetilde{x}.$$

Then, for every $t \in [s]$,

$$\frac{(B\widetilde{x})_t}{\sqrt{w(t)}} = \Big(2A_t + (3-\rho)\eta_t + \frac{\eta_t^2}{\rho-1}\Big)x_t + \frac{\eta_t}{\rho-1}U_t(x) + \frac{\beta_t}{(\rho-1)w(t)}\sum_{u\in DC^t} w(u)\big(U_u(x) + \eta_u x_u\big).$$

In particular, for $\rho = 5/2$ and $x \in \mathbb{R}^s_{\geq 0}$,

$$\frac{(B\widetilde{x})_t}{\sqrt{w(t)}} \leq \Big(2A_t + \tfrac{1}{2}g_t + \tfrac{2}{3}g_t^2\Big)x_t + \tfrac{2}{3}g_t U_t(x) + \frac{2\beta_t}{3w(t)}\sum_{u\in DC^t} w(u)U_u(x) + \frac{2\beta_t}{3w(t)}\sum_{u\in DC^t} w(u)g_u x_u.$$

Proof. This is obtained by expanding Equation (35), exactly as in the proof of Lemma 6.3. □

Lemma 6.10. *Let*

$$h_t \stackrel{def}{=} (1+11g_t)^{-2/3}, \qquad \widetilde{h}_t \stackrel{def}{=} \sqrt{w(t)}\, h_t.$$

Then

$$\lambda_{\max}(B) \leq \max_{1\leq x\leq \tau} F(x) < \frac{12}{5}.$$

Proof. Since B is symmetric and entrywise nonnegative, Collatz–Wielandt Formula gives

$$\lambda_{\max}(B) \leq \max_{t\in[s]} \frac{(B\widetilde{h})_t}{\widetilde{h}_t}.$$

The three inequalities used in the proof of Lemma 6.4 are unchanged, except that $|DC^t|$ is replaced by $M_t/w(t)$, which is exactly absorbed by the definition of A_t. Hence the same functions $G(g)$ and $F(x)$ are obtained, and the final inequality follows from Lemma 6.6. □

We can now conclude exactly as in the unweighted case:

$$\begin{aligned}
&\sum_{t\in[s]} w(t)\big(\alpha^*_t - (\rho-1)R_t^2\big) - \hat{f} \\
&\quad\leq \frac{5}{2}\sum_{t\in[s]} w(t)a_t^2 + Q^w_{5/2}(a) && \text{(Lemma 6.8)} \\
&\quad\leq \frac{5}{2}\sum_{t\in[s]} w(t)a_t^2 + \lambda_{\max}(B)\sum_{t\in[s]} w(t)a_t^2 && \text{(Lemma 6.9)} \\
&\quad\leq 4.9\sum_{t\in[s]} w(t)a_t^2. && \text{(Lemma 6.10)}
\end{aligned}$$

This proves Lemma 4.12.

7 Log-Adaptive Algorithm for Weighted Facility Location

Let $\varepsilon \in (0, 10^{-100})$ be a sufficiently small constant. We choose $\delta = \varepsilon/2$. In weighted facility location, each client $j \in D$ has a weight $w(j) > 0$ and the objective is to minimize $\sum_{j \in D} w(j) \cdot d^2(j, S) + |S|f$.

We present the modified greedy algorithm for weighted facility location that works in $O(\log n/\varepsilon^3)$ phases.

Definition 7.1 (openable)**.** *A facility i is* openable *with respect to the algorithm's state (α, S, θ) if there are increased dual values $(\tau_j)_{j \in A}$ of active clients that satisfy the following conditions:*

- *Only nearby clients are increased: $\tau_j = \alpha_j$ for every $j \in A \setminus B(i, \sqrt{\varepsilon\theta})$.*
- *Nearby clients are only increased slightly: $\alpha_j \le \tau_j \le \min\{(1-\delta)d^2(j, S), (1+\varepsilon^3)\theta\}$ for every $j \in A \cap B(i, \sqrt{\varepsilon}\theta)$.*
- *Facility i is paid for:*

$$(1-\delta) \sum_{j \in DC} w(j) \cdot [d^2(j, S) - d^2(j, i)]^+ + \sum_{j \in IC} w(j) \cdot [\alpha_j - \rho(1-\delta)d^2(j, i)]^+ + \sum_{j \in A} w(j) \cdot [\tau_j - \rho(1-\delta)d^2(j, i)]^+ \ge \hat{f}.$$

- *Dual feasibility: for every facility i_0:*

$$\sum_{j \in DC} w(j) \cdot [(1-\delta)d^2(j, S) - d^2(j, i_0)]^+ + \sum_{j \in IC} w(j) \cdot [\alpha_j - \rho d^2(j, i_0)]^+ + \sum_{j \in A} w(j) \cdot [\tau_j - \rho d^2(j, i_0)]^+ \le \hat{f}.$$

The openability can be checked by solving a linear program.

Lemma 7.2. *There is a polynomial-time algorithm that, given the state of the algorithm and a facility i, either outputs $(\tau_j)_{j \in A}$ that satisfy all conditions of openability or certifies that i is not openable.*

Proof. Fix i and $(\alpha, S, A, IC, DC, \theta)$ and consider a linear program with variables $(\tau_j)_{j \in A}$. We claim that each of the four bullets in Definition 7.1 can be expressed as linear constraints. The first two bullets are immediate. Indeed, as $\tau_j = \alpha_j$ for every $j \in A \setminus B(i, \sqrt{\varepsilon\theta})$, we can treat them as constants and only have $(\tau_j)_{j \in A \cap B(i, \sqrt{\varepsilon\theta})}$ as the true variables.

Then, the third bullet,

$$(1-\delta) \sum_{j \in DC} w(j) \cdot [d^2(j, S) - d^2(j, i)]^+ + \sum_{j \in IC} w(j) \cdot [\alpha_j - \rho(1-\delta)d^2(j, i)]^+ + \sum_{j \in A} w(j) \cdot [\tau_j - \rho(1-\delta)d^2(j, i)]^+ \ge \hat{f}.$$

becomes a linear inequality in the true variables, since the only terms involving the true variables are $w(j) \cdot [\tau_j - \rho(1-\delta)d^2(j, i)]^+$ for $j \in A \cap B(i, \sqrt{\varepsilon\theta})$. For those j, $[\tau_j - \rho(1-\delta)d^2(j, i)]^+$ is

indeed equal to $(\tau_j - \rho(1-\delta)d^2(j,i))$ (without $[\cdot]^+$) because ε is sufficiently small, $d^2(j,i) \le \varepsilon\theta$ and $\tau \ge \theta$.

Finally, for the fourth bullet, we can use an additional variable $z'_{i_0,j} \ge 0$ for each $j \in D$. We add the constraint $z'_{i_0,j} \ge (1-\delta)d^2(j,S) - d^2(j,i_0)$ for $j \in DC$, the constraint $z'_{i_0,j} \ge \alpha_j - \rho d^2(j,i_0)$ for $j \in IC$ and the constraint $z'_{i_0,j} \ge \tau_j - \rho d^2(j,i_0)$ for $j \in A$. In the linear program, we can rewrite the corresponding fourth bullet as

$$\sum_{j \in D} w(j) z'_{i_0,j} \le \hat{f} \ .$$

□

Algorithm 2 is our log-adaptive algorithm. It works in $O(\log n/\varepsilon^3)$ phases where, in the ith phase, every active client j has $\alpha_j = (1+\varepsilon^3)^i = \theta$ and the algorithm opens openable facilities in an arbitrary order by increasing α_j's to τ_j's. However, when i is open, the only scenario j with $\tau_j > \alpha_j$ is when $j \in B(i, \sqrt{\varepsilon\theta})$, so j becomes immediately put into DC after i is open. Unlike Algorithm 1, we don't lower the α-values for clients in IC immediately after a facility is opened. Instead, we only lower the α-values in the second stage of each phase.

Algorithm 2. Parameters: $\rho, \Gamma > 1, \varepsilon \in (0, 10^{-100})$; $\hat{f} = (\Gamma + 201\sqrt{\varepsilon}) \cdot f$ where f is the uniform facility opening cost.

Initialization: Set $\theta = 1$, $S = \emptyset$, $A = D$, $DC = \emptyset$, $IC = \emptyset$, and $\alpha_j = \theta$ for every $j \in D$. While $A \neq \emptyset$:

1. **While** there is an unopened facility i that is openable:
   (a) Compute $(\tau_j)_{j \in A}$ so that i is openable with $(\tau_j)_{j \in A}$.
   (b) Add i to S. Set $\alpha_j \leftarrow \tau_j$ for every $j \in A \cap B(i, \sqrt{\varepsilon\theta})$.
   (c) For every $j \in A \cup IC$:
   - If $\alpha_j \ge \rho(1-\delta)d^2(j,i)$, move j to DC and set $R_j = \sqrt{1-\delta}d(j,i)$.
   - Otherwise, if $j \in A$ and $\alpha_j \ge (1-\delta)d^2(j,i)$, move j to IC.
2. Perform the following step and move to the next phase:
   - For every $j \in IC$, set $\alpha_j \leftarrow (1-\delta)d^2(j,S)$.
   - For every remaining $j \in A$, set $\alpha_j \leftarrow \min\{(1+\varepsilon^3)\theta, (1-\delta)d^2(j,S)\}$ and move it to IC when $\alpha_j = (1-\delta)d^2(j,S)$.
   - Update $\theta \leftarrow (1+\varepsilon^3)\theta$.

For each client $j \in D$, we explicitly track its first direct-connection cost, R_j. We set $R_j = \sqrt{1-\delta}d(j,i^*)$ where i^* is the facility that caused j to be moved to DC for the first time. For a client that remains only indirectly connected or active, we set $R_j = 0$.

In the rest of this subsection, let $(\alpha^*_j)_{j \in D}$ be the final dual solution produced by the algorithm.

7.1 Dual Payment

Lemma 7.3. *At the end of the log-adaptive algorithm, the final dual solution α^* satisfies:*

$$\sum_{j\in D} w(j)\cdot(\alpha_j^* - (\rho-1)R_j^2) \geq (1-\delta)\sum_{j\in D} w(j)\cdot d^2(j,S) + \sum_{i\in S}\hat{f}$$

Proof. First, let us show that it suffices to prove that at the end of the algorithm, we have

$$\sum_{j\in DC} w(j)\cdot\alpha_j \geq \sum_{j\in DC} w(j)\cdot\left[(1-\delta)d^2(j,S) + (\rho-1)R_j^2\right] + \sum_{i\in S}\hat{f}$$

Indeed, at the end of the algorithm, $A=\emptyset$. For every $j\in IC$, we have $\alpha_j \geq (1-\delta)d^2(j,S)$ by the construction of the update steps in Stage 2. Because $R_j=0$ for $j\in IC$, their contribution naturally satisfies the required bound. Therefore, proving the DC client inequality implies the lemma.

To this end, we prove by induction that at any point in the execution of the algorithm, the DC invariant holds. The base case is at the beginning of the algorithm, where $DC=\emptyset$ and $S=\emptyset$. Next, we show that this invariant is preserved in each stage of the algorithm.

Stage 1. In each step of this stage, we open a facility i, i.e., adding it to S. Suppose that $(\alpha, S, A, IC, DC, \theta)$ denotes the state right before opening i. Let A' and IC' be the clients that are moved to DC from A and IC respectively by the opening of i. This means that $A' = \{j\in A : \tau_j \geq \rho(1-\delta)d^2(j,i)\}$ and $IC' = \{j\in IC : \alpha_j \geq \rho(1-\delta)d^2(j,i)\}$.

For these newly directly connected clients, their initial direct-connection parameter is exactly $R_j = \sqrt{1-\delta}d(j,i)$. Therefore, their combined contribution to the right-hand side of our invariant is $w(j)\cdot((1-\delta)d^2(j,i)+(\rho-1)R_j^2) = w(j)\cdot((1-\delta)d^2(j,i)+(\rho-1)(1-\delta)d^2(j,i))$. Notice that since $\rho>1$ and $\delta>0$, we can bound this contribution:

$$\begin{aligned} w(j)\cdot((1-\delta)d^2(j,i)+(\rho-1)(1-\delta)d^2(j,i)) &= w(j)(1-\delta)d^2(j,i)[1+(\rho-1)] \\ &= w(j)\rho(1-\delta)d^2(j,i) \end{aligned}$$

Let $X = \{j\in DC : d^2(j,i) < d^2(j,S)\}$ denote the clients in DC whose distance to i is smaller than their distance to S. For clients in X, their fixed penalty R_j does not change, meaning the change in cost on the right-hand side is strictly driven by the distance dropping to $d^2(j,i)$.

Using the bound established above, the change of cost of the right-hand side is at most:

$$\hat{f} + \sum_{j\in A'\cup IC'} w(j)\cdot\rho(1-\delta)d^2(j,i) - \sum_{j\in X} w(j)\cdot(1-\delta)(d^2(j,S)-d^2(j,i))$$

Since i is fully paid according to the third bullet of the openability condition, we also have the capacity bound:

$$\hat{f} \leq \sum_{j\in A'} w(j)\cdot(\tau_j-\rho(1-\delta)d^2(j,i)) + \sum_{j\in IC'} w(j)\cdot(\alpha_j-\rho(1-\delta)d^2(j,i)) + \sum_{j\in X} w(j)\cdot(1-\delta)(d^2(j,S)-d^2(j,i))$$

Substituting this capacity bound back into our change of cost, the distance and capacity terms cancel out, and we get that the change of the right-hand side is at most:

$$\sum_{j\in A'} w(j)\cdot\tau_j + \sum_{j\in IC'} w(j)\cdot\alpha_j$$

Because we do not change the α-value after a client is moved to DC, this is exactly the change of the left-hand side of our invariant. Thus, the inequality holds through Stage 1.

Stage 2. No facility is open at this stage, and no client is added to DC either. Because DC is unaltered and no new $\hat{f}$ is introduced, the invariant remains trivially preserved through the continuous updates of A and IC clients. □

7.2 Dual Feasibility

Lemma 7.4. *We have for any $i \in \mathcal{F}$,*

$$\sum_{j \in D} w(j) \cdot [\alpha_j^* - (\rho - 1)R_j^2 - (\Gamma + 201\sqrt{\varepsilon})d^2(j,i)]^+ \leq \hat{f}, \tag{36}$$

where

- *if d is a (general) metric, we choose $\rho = 2.5, \Gamma = 4.9$; or*
- *if d is a Euclidean metric, we choose $\rho = 2, \Gamma = 3 + \ln(2)$.*

Claim 7.5 (No over-bidding)**.** *At any point in the algorithm, for every facility $i_0 \in \mathcal{F}$, we have*

$$\sum_{j \in DC} w(j) \cdot [(1-\delta)d^2(j,S) - d^2(j,i_0)]^+ + \sum_{j \in A \cup IC} w(j) \cdot [\alpha_j - \rho d^2(j,i_0)]^+ \leq \hat{f}.$$

Proof. Consider an arbitrary facility i_0.

Stage 1: Suppose that a facility i' is opened with dual increases $(\tau_j)_{j \in A}$. Because i' is openable, the proposed state satisfies the fourth bullet of Definition 7.1 (dual feasibility) for any $i_0 \in \mathcal{F}$:

$$\sum_{j \in DC} w(j) \cdot [(1-\delta)d^2(j,S) - d^2(j,i_0)]^+ + \sum_{j \in IC} w(j) \cdot [\alpha_j - \rho d^2(j,i_0)]^+ + \sum_{j \in A} w(j) \cdot [\tau_j - \rho d^2(j,i_0)]^+ \leq \hat{f}.$$

Let us analyze how the left-hand side of our actual state constraint differs from this openability bound after i' is added to S and clients are processed. First, observe that for every active client $j \in A$, its updated dual value effectively becomes τ_j. This is because Step 1(b) explicitly sets $\alpha_j \leftarrow \tau_j$ for $j \in A \cap B(i', \sqrt{\varepsilon\theta})$, and for active clients outside this ball, the first openability condition guarantees $\tau_j = \alpha_j$. Thus, going into Step 1(c), the duals are τ_j for A and remain α_j for IC.

For a client j that was already directly connected (DC) before opening i', its contribution can only decrease, since the distance $d^2(j, S \cup \{i'\}) \leq d^2(j,S)$.

For a client j that was active (A) or indirectly connected (IC) and now moves to DC, it does so precisely because its updated dual value satisfies the condition in Step 1(c): $\tau_j \geq \rho(1-\delta)d^2(j, S \cup \{i'\})$ (if it was from A) or $\alpha_j \geq \rho(1-\delta)d^2(j, S \cup \{i'\})$ (if it was from IC). Since $\rho > 1$ and the function $[x]^+$ is non-negative and non-decreasing, we can strictly bound its new actual contribution. For a client moving from A:

$$[(1-\delta)d^2(j, S \cup \{i'\}) - d^2(j,i_0)]^+ \leq \left[\frac{\tau_j}{\rho} - d^2(j,i_0)\right]^+ \leq \left[\tau_j - \rho d^2(j,i_0)\right]^+.$$

The identical algebraic bound holds for a client moving from IC using α_j. In both cases, the client's new connection contribution is strictly bounded by its corresponding dual term in the openability bound.

Finally, for the active or indirectly connected clients that do not move to DC, their actual dual values are exactly τ_j and α_j, respectively. Whether an active client remains in A or moves to IC, its contribution is evaluated using its dual variable, resulting in $w(j)[\tau_j - \rho d^2(j, i_0)]^+$. This perfectly matches its term in the openability bound. Therefore, every term in the updated state is less than or equal to its corresponding term in the openability bound, guaranteeing the total sum remains $\leq \hat{f}$.

Stage 2: In the second stage, we have the following steps: decreasing the α-values of clients in IC to $(1-\delta)d^2(j, S)$, and increasing the α-values of clients in A to $\min\{(1+\varepsilon^3)\theta, (1-\delta)d^2(j, S)\}$.

The first step can only decrease the left-hand side of the claim. Hence, it suffices to show that the continuous increase in the second step does not violate the claim. Assume towards contradiction that at the end of the phase, if we increase α_j of every $j \in A$ to $\min\{(1+\varepsilon^3)\theta, (1-\delta)d^2(j, S)\}$, the claim is violated for a non-empty subset of facilities F'.

We select a "minimal" such counter-example: let $\tau' \leq (1+\varepsilon^3)\theta$ be the smallest value such that if we set $\alpha'_j := \min\{\tau', (1-\delta)d^2(j, S)\}$, it satisfies

$$\sum_{j\in A} w(j)[\alpha'_j - \rho d^2(i,j)]^+ + \sum_{j\in IC} w(j)[\alpha_j - \rho d^2(i,j)]^+ + \sum_{j\in DC} w(j)[(1-\delta)d^2(j,S) - d^2(i,j)]^+ = \hat{f}$$

for some $i \in F'$. We remark that $\tau' \geq \theta$ since the constraint was strictly satisfied at the end of Phase 1. Furthermore, $i \notin S$ is not yet open; if it were open, there would be no active client $j \in A$ for which $\alpha'_j - \rho d^2(i, j)$ is strictly positive (since $\alpha'_j \leq (1-\delta)d^2(j, S) \leq (1-\delta)d^2(j, i) < \rho d^2(j, i)$). Hence, the increase in α-values cannot cause i to violate the constraint if it were already opened, which contradicts $i \in F'$.

We now show that this facility i, combined with dual values $\tau_j = \alpha'_j$ for $j \in A \cap B(i, \sqrt{\varepsilon\theta})$ and $\tau_j = \alpha_j$ for $j \in A \setminus B(i, \sqrt{\varepsilon\theta})$, satisfies the conditions of openability. In other words, i is an openable facility that was not yet opened, which contradicts the proper completion of the first stage's while-loop.

We verify the conditions of openability one-by-one. The first two bullets (only nearby clients increased, and only increased slightly) are directly satisfied by our choice of $\tau_j \leq \tau'$ and the definition of α'_j. For the third bullet (facility i is paid for):

$$\begin{aligned}
&\sum_{j\in A} w(j)\cdot[\tau_j - \rho(1-\delta)d^2(i,j)]^+ + \sum_{j\in IC} w(j)\cdot[\alpha_j - \rho(1-\delta)d^2(i,j)]^+ \\
&+ \sum_{j\in DC} w(j)\cdot[(1-\delta)d^2(j,S) - (1-\delta)d^2(i,j)]^+ \\
&\geq \sum_{j\in A\cap B(i,\sqrt{\varepsilon\theta})} w(j)\cdot[\alpha'_j - \rho d^2(i,j)]^+ + \sum_{j\in A\setminus B(i,\sqrt{\varepsilon\theta})} w(j)\cdot[\alpha'_j - \rho d^2(i,j)]^+ \\
&+ \sum_{j\in IC} w(j)\cdot[\alpha_j - \rho d^2(i,j)]^+ + \sum_{j\in DC} w(j)\cdot[(1-\delta)d^2(j,S) - d^2(i,j)]^+ \\
&\geq \hat{f}.
\end{aligned}$$

Here, for the subset outside the ball $A \setminus B(i, \sqrt{\varepsilon\theta})$, we used the fact that $\alpha_j - \rho(1-\delta)d^2(i, j) \geq (1+\varepsilon^3)\alpha_j - \rho d^2(i, j) \geq \alpha'_j - \rho d^2(i, j)$ when $d^2(i, j) \geq \varepsilon\theta$ (recalling that $\alpha_j = \theta$ for all active clients).

It remains to verify the fourth bullet of openability (dual feasibility). For any facility $i_0 \in \mathcal{F}$, we evaluate the dual feasibility constraint using our constructed $(\tau_j)_{j\in A}$. Because

$\tau_j \le \alpha'_j$ for all $j \in A$, and the function $[x]^+$ is non-decreasing, we can explicitly bound the left-hand side of the dual feasibility constraint:

$$\sum_{j\in DC} w(j)[(1-\delta)d^2(j,S) - d^2(j,i_0)]^+ + \sum_{j\in IC} w(j)[\alpha_j - \rho d^2(j,i_0)]^+ + \sum_{j\in A} w(j)[\tau_j - \rho d^2(j,i_0)]^+$$
$$\le \sum_{j\in DC} w(j)[(1-\delta)d^2(j,S) - d^2(j,i_0)]^+ + \sum_{j\in IC} w(j)[\alpha_j - \rho d^2(j,i_0)]^+ + \sum_{j\in A} w(j)[\alpha'_j - \rho d^2(j,i_0)]^+$$
$$\le \hat{f}.$$

The first inequality holds by direct substitution of the upper bound $\tau_j \le \alpha'_j$. The second inequality follows strictly from the minimality of τ': by definition, τ' was chosen as the *smallest* value at which the total sum (evaluated with α'_j) reaches $\hat{f}$ for *some* facility ($i \in F'$). Therefore, for any arbitrary facility i_0, this sum cannot strictly exceed $\hat{f}$. Since all four conditions hold, i is openable, which is a contradiction. □

Next, we fix a facility i and show that the dual-feasibility for i is satisfied at the end of the algorithm. We say that facility i is frozen if there exists some open facility i_0 such that

$$d(i,i_0) \le 2\sqrt{\varepsilon\theta}\ .$$

Modified α-values. For analysis purposes, we will introduce a new modification of the α-values for each client j in terms of the fixed facility i. We will denote this modified α-value as α^c_j. This modified α^c-value will be used to write the analog of the constraints in Lemmas 4.9 and 4.10.

The modifications happen only when we open a facility i_0 that does not freeze i, i.e., when $d(i,i_0) > 2\sqrt{\varepsilon\theta}$. More specifically, at the stage with time θ, if we open such a facility i_0 that does not freeze i, we keep α^c_j as the value before opening i_0. That is, we don't increase α^c_j to τ_j if $j \in B(i_0, \sqrt{\varepsilon\theta})$ (step 1b) for such i_0. In other steps (step 1b for a facility i_0 that freezes i or step 2), we still modify α^c_j in the same way as we modify α_j.

In the rest of this subsection, we will consider α^c instead of α^* in writing the constraints. To ensure that the α^*-value can ensure dual feasibility, we will show that α^c_j and α^*_j are close to each other for every client j.

Lemma 7.6. *For any client $j \in D^*$, we have $\alpha^*_j - \alpha^c_j \le \varepsilon^2 \cdot d^2(j,i)$.*

Proof. According to the definition of α^c_j, the only case where $\alpha^c_j \ne \alpha^*_j$ is if at some time θ, client j becomes directly connected to a facility i_0 that does not freeze i. Suppose that α_j is the α-value of j before opening i_0 and τ_j is the α-value of j after opening i_0. Because we will no longer change the α-value of a client if it is in DC, we have $\alpha^c_j = \alpha_j$ and $\alpha^*_j = \tau_j$. Note that $\tau_j - \alpha_j \le \varepsilon^3\theta$ according to Definition 7.1. On the other hand, if $\tau_j > \alpha_j$, we have $d(j,i_0) \le \sqrt{\varepsilon\theta}$. Since i_0 does not freeze i, we know $d(i,i_0) > 2\sqrt{\varepsilon\theta}$. By the triangle inequality, we have $d(j,i) \ge d(i,i_0) - d(j,i_0) > \sqrt{\varepsilon\theta}$. Squaring both sides implies $d^2(j,i) > \varepsilon\theta$, which rearranges to $\theta < \frac{1}{\varepsilon}d^2(j,i)$. Substituting this into our dual difference yields $\alpha^*_j - \alpha^c_j = \tau_j - \alpha_j \le \varepsilon^3\theta < \varepsilon^2 d^2(j,i)$. □

Now, we assume that

$$D^* = \{j \in D : \alpha^c_j - (\rho-1)R^2_j > \Gamma d^2(j,i)\}\ . \tag{37}$$

Then, it suffices to show that

$$\sum_{j\in D^*} w(j)\cdot(\alpha_j^c-(\rho-1)R_j^2)\le \hat{f}+(\Gamma+200\sqrt{\varepsilon})\cdot\sum_{j\in D^*} w(j)\cdot d^2(j,i)\ .$$

For simplicity, let us number the clients in D^* from 1 to $s=|D^*|$ in the order they become inactive, breaking ties by order of increasing α^c- value. We will use D^* and $[s]$ interchangeably.

Establishing the constraints. Next, for each client $t\in[s]$:

- If t becomes inactive strictly before i is frozen, we define DC^t as the set of clients in $[t-1]$ that are directly connected at the beginning of the stage t becomes inactive, $IC^t=[t-1]\setminus DC^t$, and S^t as the set of opened facilities at the beginning of the stage t becomes inactive.
- Else, we define DC^t as the set of clients in $[t-1]$ that are directly connected right before t becomes inactive, $IC^t=[t-1]\setminus DC^t$, and S^t as the set of opened facilities right before t becomes inactive.

Claim 7.7. *For any $t\in D^*$ that becomes inactive strictly before i becomes frozen,*

$$\sum_{j\in DC^t} w(j)[(1-\delta)d^2(j,S^t)-d^2(j,i)]^+ + \sum_{j\in IC^t} w(j)[\alpha_j^c-\rho d^2(j,i)]^+ + \sum_{j\ge t} w(j)[\alpha_t^c-\rho d^2(j,i)]^+ \le \hat{f}.$$

Proof. Let us analyze the situation depending on the stage in which t becomes inactive. Since the claim assumes t becomes inactive strictly before i becomes frozen, any facility i_0 opened at or before the time t becomes inactive cannot freeze i.

Stage 1. Suppose t becomes inactive because of the opening of some facility i_0. Because t becomes inactive before i is frozen, i_0 does not freeze i. By the definition of the modified duals, α^c does not incorporate the Step 1b increases associated with opening a non-freezing facility i_0. Specifically, for any client j, its modified value α_j^c is bounded by its actual dual value α_j at the beginning of the stage i_0 is opened.

At the beginning of the stage where i_0 is opened, let α denote the current dual values. We have $\alpha_j=\theta$ for all active clients, including t and all $j>t$. Therefore, the modified dual for t is exactly $\alpha_t^c=\theta$. By applying Claim 7.5 (no over-bidding) for facility i at the beginning of the stage i_0 is opened, we establish:

$$\sum_{j\in DC^t} w(j)[(1-\delta)d^2(j,S^t)-d^2(j,i)]^+ + \sum_{j\in IC^t} w(j)[\alpha_j-\rho d^2(j,i)]^+ + \sum_{j\ge t} w(j)[\alpha_j-\rho d^2(j,i)]^+ \le \hat{f}.$$

For any $j\in IC^t$, the modified dual value α_j^c never exceeds the true dual value α_j at the beginning of the stage i_0 was opened, so $\alpha_j^c\le\alpha_j$. For the active clients $j\ge t$, we have $\alpha_j=\theta=\alpha_t^c$. Substituting these into the inequality, and noting that the function $[v]^+$ is non-decreasing with respect to v, we obtain the desired bound:

$$\sum_{j\in DC^t} w(j)[(1-\delta)d^2(j,S^t)-d^2(j,i)]^+ + \sum_{j\in IC^t} w(j)[\alpha_j^c-\rho d^2(j,i)]^+ + \sum_{j\ge t} w(j)[\alpha_t^c-\rho d^2(j,i)]^+ \le \hat{f}.$$

Stage 2. Suppose that t becomes inactive due to the continuous update of the α-values in Stage 2 (at the end of a phase). In Stage 2, the modified duals α^c track the actual duals α exactly, meaning $\alpha_t^c = \alpha_t \leq (1+\varepsilon^3)\theta$.

By Claim 7.5, the over-bidding constraint is satisfied for facility i at the exact moment t becomes inactive:

$$\sum_{j \in DC} w(j)[(1-\delta)d^2(j,S) - d^2(j,i)]^+ + \sum_{j \in A \cup IC} w(j)[\alpha_j - \rho d^2(j,i)]^+ \leq \hat{f}.$$

Notice that Stage 2 does not alter the sets DC and $IC \cup A$. Therefore, at this moment, $DC^t \subseteq DC$ and $IC^t \subseteq IC \cup A$. We compare the contributions of each client in this inequality to the terms in our claim:

- For $j \in DC^t \subseteq DC$, the contribution is identically $[(1-\delta)d^2(j,S^t) - d^2(j,i)]^+$.
- For $j \in IC^t$, we have $\alpha_j^c \leq \alpha_j$ since dual values are non-increasing once a client is in IC, and our modification α^c strictly lowers or matches α. Thus, the term for j is upper bounded by the corresponding term in the over-bidding claim.
- For $j > t$, these clients either remain in A, or becomes inactive at the same stage as t but have $\alpha_j^c \geq \alpha_t^c$. In the first case, we have $\alpha_t = (1+\varepsilon^3)\theta$, and thus $\alpha_t^c \leq \alpha_t \leq \alpha_j$ because α^c-value is non-increasing after becoming inactive. In the second case, we have $\alpha_t^c \leq \alpha_j^c \leq \alpha_j$. Thus, $[\alpha_t^c - \rho d^2(j,i)]^+ \leq [\alpha_j - \rho d^2(j,i)]^+$.

Summing these bounds confirms the inequality holds. □

Claim 7.8. *If i is frozen by i_0 with $(\tau_j)_{j \in A}$, then, for every $j \in D^* \cap A$, j becomes inactive when opening i_0 and $\alpha_j^c \leq \tau_j$.*

Proof. Consider $j \in A$. First, note that if $\tau_j \geq (1-\delta)d^2(j,i_0)$, then j is removed from A when i_0 is opened (moving to either IC or DC), and so $\alpha_j^c \leq \tau_j$ since α-values are non-increasing after a client becomes inactive.

In the other case, when $\alpha_j \leq \tau_j < (1-\delta)d^2(j,i_0)$, where α_j is the α-value of j before i_0 is open, we show that $j \notin D^*$. Consider the (future) time right after j is removed from the active clients. The α-value of j then (which upper bounds the final α_j^c), cannot be strictly greater than $(1-\delta)d^2(j,i_0)$ whether it is increased in stage 1 or in stage 2, since we have $i_0 \in S$ at that time. By the metric triangle inequality:

$$d(i,j) \geq d(j,i_0) - d(i_0,i) = \left(1 - \frac{d(i_0,i)}{\sqrt{\theta}} \cdot \frac{\sqrt{\theta}}{d(j,i_0)}\right) d(j,i_0)$$

where θ is the value when i_0 was open. As i_0 freezes i, we have $d(i_0,i) \leq 2\sqrt{\varepsilon\theta}$, and we know $\theta = \alpha_j < (1-\delta)d^2(j,i_0)$. Plugging in those bounds to the above inequality yields:

$$d(i,j) > \left(1 - 2\sqrt{\varepsilon(1-\delta)}\right) d(j,i_0) \ .$$

Because $\varepsilon \in (0, 10^{-100})$ is sufficiently small, the term $2\sqrt{\varepsilon(1-\delta)}$ is strictly less than 1, meaning the distance remains strictly positive. Squaring both sides gives:

$$d^2(i,j) > \left(1 - 2\sqrt{\varepsilon(1-\delta)}\right)^2 d^2(j,i_0) \ .$$

Since $\Gamma > 2$, we have $\Gamma\left(1 - 2\sqrt{\varepsilon(1-\delta)}\right)^2 > 1 > 1 - \delta$. Thus:

$$\Gamma d^2(i,j) > (1-\delta)d^2(j,i_0) \geq \alpha_j^c \ .$$

Hence, $\alpha_j^c < \Gamma d^2(i,j)$, which implies j cannot be in D^*. □

Claim 7.9. *For any $t \in D^*$,*

$$\sum_{j \in DC^t} w(j)((1-\delta)d^2(j,S^t) - d^2(j,i)) + \sum_{j \in IC^t} w(j)(\alpha_j^c - \rho d^2(j,i)) + \sum_{j \geq t} w(j)(\alpha_t^c - \rho d^2(j,i)) \leq \hat{f}.$$

Proof. Consider the point in the execution at which t becomes inactive. As established by Claim 7.8, if a facility i_0 freezes i, every active client $j \in D^* \cap A$ becomes inactive during the exact step i_0 is opened. Therefore, no client in D^* can become inactive *after* i is frozen. This leaves us with exactly two exhaustive cases for $t \in D^*$:

Case 1: t becomes inactive strictly before i is frozen. By Claim 7.7,

$$\sum_{j \in DC^t} w(j)[(1-\delta)d^2(j,S^t) - d^2(j,i)]^+ + \sum_{j \in IC^t} w(j)[\alpha_j^c - \rho d^2(j,i)]^+ + \sum_{j \geq t} w(j)[\alpha_t^c - \rho d^2(j,i)]^+ \leq \hat{f}.$$

Dropping the $[\cdot]^+$ completes the proof for this case.

Case 2: t becomes inactive exactly when i is frozen. Suppose t becomes inactive due to the opening of a facility i_0 that freezes i. Because all clients in $D^* \cap A$ become inactive at this step (Claim 7.8), and our ordering of $1, \ldots, s$ in D^* is determined by the time they become inactive, it must be that all clients $j \geq t$ in D^* become inactive simultaneously at this step. Furthermore, because ties are broken by order of increasing α^c-value, we are guaranteed that $\alpha_t^c \leq \alpha_j^c$ for all $j \geq t$.

Now, consider the fourth bullet of openability (dual feasibility) for i_0 evaluated at the fixed facility i, just before i_0 is opened:

$$\sum_{j \in DC} w(j)[(1-\delta)d^2(j,S) - d^2(j,i)]^+ + \sum_{j \in IC} w(j)[\alpha_j - \rho d^2(j,i)]^+ + \sum_{j \in A} w(j)[\tau_j - \rho d^2(j,i)]^+ \leq \hat{f}.$$

Thus, letting A_{before} and IC_{before} be the sets A and IC just before i_0 is opened, we have exactly $S = S^t$, $DC^t \subseteq DC$, and $IC^t = (IC^t \cap IC_{before}) \cup (IC^t \cap A_{before})$. Note also that $t \in IC^t \cap A_{before}$, and all $j > t$ belong to A_{before}.

We drop the non-negative $[\cdot]^+$ operators from the openability constraint and restrict the sums to our subsets of interest to establish a strict lower bound on the left-hand side:

- For $j \in DC^t \subseteq DC$, we drop the positive part. Because $S = S^t$, we exactingly have: $[(1-\delta)d^2(j,S) - d^2(j,i)]^+ \geq (1-\delta)d^2(j,S^t) - d^2(j,i)$.
- For $j \in IC^t \cap IC_{before}$, the modified dual never exceeds the true dual prior to freezing, so $\alpha_j^c \leq \alpha_j$. Thus: $[\alpha_j - \rho d^2(j,i)]^+ \geq \alpha_j - \rho d^2(j,i) \geq \alpha_j^c - \rho d^2(j,i)$.
- For $j \in IC^t \cap A_{before}$, these clients are active and in D^*. By Claim 7.8, we have $\alpha_j^c \leq \tau_j$. Evaluated in the summation over A, we have: $[\tau_j - \rho d^2(j,i)]^+ \geq \tau_j - \rho d^2(j,i) \geq \alpha_j^c - \rho d^2(j,i)$. (Notice that combining this with the previous bullet perfectly reconstructs the $\alpha_j^c - \rho d^2(j,i)$ term for *all* $j \in IC^t$).

- For $j \geq t$, these clients are in $D^* \cap A_{before}$. By Claim 7.8, $\alpha_j^c \leq \tau_j$. Applying our tie-breaking rule, $\alpha_t^c \leq \alpha_j^c \leq \tau_j$. Evaluated in the A sum, we have: $[\tau_j - \rho d^2(j,i)]^+ \geq \tau_j - \rho d^2(j,i) \geq \alpha_t^c - \rho d^2(j,i)$.

Because all subsets are disjoint and strictly contained within the original openability sums, summing these lower bounds yields:

$$\sum_{j \in DC^t} w(j)((1-\delta)d^2(j,S^t) - d^2(j,i)) + \sum_{j \in IC^t} w(j)(\alpha_j^c - \rho d^2(j,i)) + \sum_{j \geq t} w(j)(\alpha_t^c - \rho d^2(j,i)) \leq \hat{f}.$$

This matches the required inequality, completely resolving the case. □

For any client t and $j \in DC^t$, define

$$r_{jt} := \sqrt{1-\delta} d(j, S^t).$$

Next, we will prove Lemma 7.4 for the general metric setting and the Euclidean setting separately based appropriate choices of ρ and Γ.

General metric

Now, we will show Lemma 7.4 for $\rho = 2.5$ and $\Gamma = 4.9$. To this end, let's show that $(IC^t), (DC^t), (\alpha_j^c), (R_j), (r_{jt})$ satisfy all the constraints of Lemma 4.12, restated below for convenience.

Lemma 4.12. *Let $\rho = 2.5$ and $\Gamma = 4.9$. Let $\hat{f}$ be any non-negative real number. Let $([s] \cup \{i\}, d)$ be a metric. Let $w : [s] \to \mathbb{R}_{>0}$ be any positive weight function. Let $(IC^t)_{t \in [s]}$ and $(DC^t)_{t \in [s]}$ be two families of sets satisfying:*

(a) For each $t \in [s]$, $DC^t \cap IC^t = \emptyset$ and $DC^t \cup IC^t = [t-1]$.

(b) For each $t \in [s-1]$, $DC^t \subseteq DC^{t+1}$.

Suppose that the variables $(\alpha_j^*)_{j \in [s]}, (R_j)_{j \in [s]}, (r_{jt})_{t \in [s], j \in DC^t}$ satisfy the following constraints

(1) For any $t \in [s], j \in DC^t$, we have $r_{jt} \leq R_j$.

(2) For each $t \in [s], j \in DC^t$, we have $\alpha_t^* \leq (r_{jt} + d(j,t))^2$.

(3) For each $t \in [s]$, we have

$$\sum_{\ell \in [t:s]} w(\ell)\alpha_t^* + \sum_{\ell \in IC^t} w(\ell)\alpha_\ell^* + \sum_{j \in DC^t} w(j) r_{jt}^2 \leq \hat{f} + \sum_{\ell \in IC^t \cup [t:s]} \rho w(\ell) d^2(\ell, i) + \sum_{j \in DC^t} w(j) d^2(j,i).$$

Then, the following holds:

$$\sum_{j \in [s]} w(j) \cdot (\alpha_j^* - (\rho - 1)R_j^2) \leq \hat{f} + \Gamma \cdot \sum_{j \in [s]} w(j) \cdot d^2(j,i) \ . \tag{9}$$

By definition of $(IC^t)_t$ and $(DC^t)_t$, the two constraints on them hold. Let $j \in DC^t$. Since R_j is the distance to the first facility that j directly connects to, and $d(j, S^t)$ is non-increasing in t, we have $r_{jt} \leq R_j$ for any $t > j$. Now, let $t \in [s]$, we have:

$$\begin{aligned}
\alpha_t^c &\leq \alpha_t^* \\
&\leq (1-\delta) d^2(t, S^t) \\
&\leq (\sqrt{1-\delta} d(t,j) + \sqrt{1-\delta} d(j, S^t))^2 \\
&\leq (r_{jt} + d(j,t))^2.
\end{aligned}$$

Claim 7.9 gives the last constraint.

Hence, by Lemma 4.12, we have:

$$\sum_{j\in[s]} w(j)\cdot(\alpha_j^c-(\rho-1)R_j^2)\le \hat{f}+\Gamma\cdot\sum_{j\in[s]} w(j)\cdot d^2(j,i).$$

This completes the proof of Lemma 7.4 for $\rho=2.5$ and $\Gamma=4.9$.

Euclidean metric

Now, we will show Lemma 7.4 for $\rho=2$, $\Gamma=3+\ln 2$, and d a Euclidean metric. Let a be the index of the last client that becomes inactive strictly before i is frozen.

First, let's prove the following claim:

Claim 7.10. *For each $t\in[a]$, $j\in DC^t$ and $\ell\in IC^t\cup\{t\}$, we have:*

$$\alpha_\ell^c\le(r_{jt}+d(j,\ell))^2.$$

Proof. Consider any client $t\in[a]$. Let P_t be the phase in which t becomes inactive. Consider the state of the algorithm at the exact beginning of phase P_t. By definition, S^t is the set of open facilities at this time.

Consider any client $\ell\in IC^t\cup\{t\}$. As $\ell\le t$, ℓ becomes inactive in phase P_t or in an earlier phase. Next, we prove that $\alpha_\ell^c\le(1-\delta)d^2(\ell,S^t)$ by discussing two cases:

- If ℓ becomes inactive strictly before phase P_t, then ℓ is indirectly connected at the beginning of phase P_t. Because of the Stage 2 continuous updates at the end of the previous phase, we have $\alpha_\ell^c\le\alpha_\ell=(1-\delta)d^2(\ell,S^t)$.
- Otherwise (i.e., if ℓ becomes inactive during phase P_t), we can prove $\alpha_\ell^c\le(1-\delta)d^2(\ell,S^t)$ by contradiction. Suppose that $\alpha_\ell^c>(1-\delta)d^2(\ell,S^t)$. Let θ be the active dual parameter at the beginning of phase P_t. Since ℓ is active at this time, $\theta=\alpha_\ell^c>(1-\delta)d^2(\ell,S^t)$. Then, ℓ would have been moved to IC during the Stage 2 update of the previous phase (when the threshold $(1+\varepsilon^3)\theta_{prev}=\theta$ exceeded the connection cost), contradicting that ℓ is still active at the beginning of phase P_t. Therefore, $\alpha_\ell^c\le\theta\le(1-\delta)d^2(\ell,S^t)$.

Because the triangle inequality holds for the Euclidean metric d, we have $d(\ell,S^t)\le d(j,S^t)+d(j,\ell)$. Taking the square root of our established bound, we get:

$$\begin{aligned}\sqrt{\alpha_\ell^c}&\le\sqrt{1-\delta}d(\ell,S^t)\\&\le\sqrt{1-\delta}(d(j,S^t)+d(j,\ell))\\&=r_{jt}+\sqrt{1-\delta}d(j,\ell)\end{aligned}$$

Since $\delta>0$, we have $\sqrt{1-\delta}<1$. Because metric distances are non-negative, replacing the scaled distance with the full distance preserves the inequality:

$$\sqrt{\alpha_\ell^c}\le r_{jt}+d(j,\ell)$$

Squaring both sides yields $\alpha_\ell^c\le(r_{jt}+d(j,\ell))^2$, completing the proof. □

We will make several cases according to how i gets frozen.

Case 1: All clients in D^* become inactive strictly before i is frozen. In this case, let's show that $(IC^t), (DC^t), (\alpha_j^c), (R_j), (r_{jt})$ satisfy all the constraints of Lemma 4.11, restated below for convenience.

Lemma 4.11. *Let $\rho = 2$. Let $\hat{f}$ be any non-negative real number. Let $([s] \cup \{i\}, d)$ be a general metric. Let $w : [s] \to \mathbb{R}_{>0}$ be any positive weight function. Let $(IC^t)_{t\in[s]}$ and $(DC^t)_{t\in[s]}$ be two families of sets satisfying:*

(a) For each $t \in [s]$, $DC^t \cap IC^t = \emptyset$ and $DC^t \cup IC^t = [t-1]$.

(b) For each $t \in [s-1]$, $DC^t \subseteq DC^{t+1}$.

Suppose that the variables $(\alpha_j^*)_{j\in[s]}, (R_j)_{j\in[s]}, (r_{jt})_{1\le j<t\le s}$ satisfy the following constraints

(1) For any $t \in [s], j \in DC^t$, we have $r_{jt} \le R_j$.

(2) For each $t \in [s], j \in DC^t$ and $\ell \in IC^t \cup \{t\}$, we have $\alpha_\ell^* \le (r_{jt} + d(j,\ell))^2$.

(3) For each $t \in [s]$, we have

$$\sum_{\ell\in[t:s]} w(\ell)\alpha_t^* + \sum_{\ell\in IC^t} w(\ell)\alpha_\ell^* + \sum_{j\in DC^t} w(j)r_{jt}^2 \le \hat{f} + \sum_{\ell\in IC^t\cup[t:s]} \rho\cdot w(\ell)d^2(\ell,i) + \sum_{j\in DC^t} w(j)d^2(j,i).$$

Then, the following holds:

$$\sum_{j\in[s]} w(j)\cdot(\alpha_j^* - (\rho-1)R_j^2) \le \hat{f} + \Gamma\cdot\sum_{j\in[s]} w(j)\cdot d^2(j,i)\ , \tag{8}$$

where $\Gamma = 5$ if d is a general metric, and $\Gamma = 3 + \ln 2$ if d is a Euclidean metric.

Since in this case $a = s$, Claim 7.10 implies constraint (2) is satisfied. Claim 7.9 implies constraint (3) is satisfied. Hence, we can apply Lemma 4.11, and get:

$$\sum_{j\in[s]} w(j)\cdot(\alpha_j^c - (\rho-1)R_j^2) \le \hat{f} + \Gamma\cdot\sum_{s\in[s]} w(j)\cdot d^2(j,i),$$

which completes the proof in this case.

Case 2: Some clients in D^* become inactive when i is frozen. In this case, we assume that i_0 freezes i. Let IC^F be the set of indirectly connected clients right before i_0 is opened, and

$$L := \{j \in IC^F : d(j,i) > \sqrt{\varepsilon\theta}\}.$$

Note that IC^F is different with IC^s in the way that IC^F does not include active clients right before i_0 is opened, i.e., $j \notin IC^F$ for any $j \in [a+1:s]$ while $IC^s = IC^F \cup [a+1:s-1]$.

We will further discuss two subcases to prove Lemma 7.4.

Subcase 2(a): $[a+1:s] \cap B(i_0, \sqrt{\varepsilon\theta}) = \emptyset$ or $\sum_{j\in L} w(j) > \varepsilon\cdot\sum_{j\in[a+1:s]} w(j)$. We redefine DC^s and IC^s as the respectively directly connected and active or indirectly connected client at the beginning of the stage s becomes inactive. First, let's bound the total increment of α_j^c when i_0 is opened.

Claim 7.11. *The total increment of α_j^c for $j \in D^*$ when i_0 is opened is at most $\varepsilon\sum_{j\in D^*} w(j)d^2(j,i)$.*

Proof. If $[a+1:s] \cap B(i_0, \sqrt{\varepsilon\theta}) = \emptyset$, or if no active client in $[a+1:s]$ strictly increases its α^c-value when i_0 is opened, the total increment for all $j \in D^*$ is exactly 0. Because weights and squared distances are non-negative, the bound $0 \le \varepsilon \sum_{j \in D^*} w(j) d^2(j,i)$ holds trivially.

Next, we consider the case where $\sum_{j \in L} w(j) > \varepsilon \cdot \sum_{j \in [a+1:s]} w(j)$.

Let $U = [a+1:s]$ denote the set of clients in D^* that become inactive exactly when i is frozen (i.e., when i_0 is opened). For any client $j \in [1:a]$, j became inactive strictly before this step. By the algorithm's rules, the dual values of inactive clients are never increased during Stage 1. Therefore, the total increment of α_j^c across all $j \in D^*$ is exactly the sum of the increments for the active clients in U.

Right before i_0 is opened, every $j \in U$ is active, meaning its dual value is exactly $\alpha_j = \theta$. When i_0 is opened, its modified dual value α_j^c is set to τ_j. By the second bullet of openability (Definition 7.1), the maximum increased value is bounded by $\tau_j \le (1+\varepsilon^3)\theta$. Thus, for every $j \in U$, the increment is bounded by $\varepsilon^3\theta$.

Summing this over all $j \in U$, the total increment for D^* is bounded by:

$$\sum_{j \in U} w(j)(\alpha_j^c - \alpha_j) \le \varepsilon^3 \theta \sum_{j \in U} w(j).$$

By our Case 2(a) assumption, we have $\sum_{j \in L} w(j) > \varepsilon \sum_{j \in U} w(j)$, which rearranges to:

$$\sum_{j \in U} w(j) < \frac{1}{\varepsilon} \sum_{j \in L} w(j).$$

Substituting this weight bound into our total increment yields:

$$\sum_{j \in U} w(j)(\alpha_j^c - \alpha_j) < \varepsilon^3 \theta \left(\frac{1}{\varepsilon} \sum_{j \in L} w(j) \right) = \varepsilon^2 \theta \sum_{j \in L} w(j).$$

Recall the definition of the subset $L = \{j \in IC^F : d(j,i) > \sqrt{\varepsilon\theta}\}$. For every client $j \in L$, we have $d^2(j,i) > \varepsilon\theta$, which algebraically rearranges to $\theta < \frac{1}{\varepsilon} d^2(j,i)$. Substituting this upper bound for θ into our increment expression, we obtain:

$$\sum_{j \in U} w(j)(\alpha_j^c - \alpha_j) < \varepsilon^2 \sum_{j \in L} w(j) \left(\frac{d^2(j,i)}{\varepsilon} \right) = \varepsilon \sum_{j \in L} w(j) d^2(j,i).$$

Finally, note that $L \subseteq IC^F \subseteq [1:a] \subset D^*$. Because client weights $w(j)$ are strictly positive and squared metric distances $d^2(j,i)$ are non-negative, the sum over the subset L is trivially bounded by the sum over the entire set D^*:

$$\varepsilon \sum_{j \in L} w(j) d^2(j,i) \le \varepsilon \sum_{j \in D^*} w(j) d^2(j,i).$$

This completes the proof. □

Apply Claim 7.5 at the beginning of the stage s becomes inactive, this lemma guarantees that

$$w(s)(\alpha_s^c - \rho d^2(s,i)) + \sum_{\ell \in IC^s} w(\ell)(\alpha_\ell^c - \rho d^2(\ell,i)) + \sum_{j \in DC^s} w(j)(r_{js}^2 - d^2(j,i)) \le \hat{f} + \varepsilon \sum_{j \in D^*} w(j) d^2(j,i).$$

Note that because we redefine IC^s, DC^s, S^s according to the sets IC, DC, S at the beginning of the stage s becomes inactive, it's easy to see that Claim 7.10 generalizes for $t = s$. Thus, we can apply Lemma 5.20 (restated, see below), replacing $\hat{f}$ with $\hat{f} + \varepsilon \sum_{j \in D^*} w(j) d^2(j, i)$ and using $L = \emptyset$ and $\varepsilon' = 0$ (the ε-parameter in the lemma). We get:

$$\sum_{j \in [s]} w(j) \cdot (\alpha_j^c - (\rho - 1) R_j^2) \leq \hat{f} + (\Gamma + \varepsilon) \cdot \sum_{j \in [s]} w(j) \cdot d^2(j, i).$$

This completes the proof in this case.

Subcase 2(b): $[a+1 : s] \cap B(i_0, \sqrt{\varepsilon\theta}) \neq \emptyset$ and $\sum_{j \in L} w(j) \leq \varepsilon \cdot \sum_{j \in [a+1:s]} w(j)$. In this case, we define DC^s and IC^s as respectively the directly connected and the active or indirectly connected client right before s becomes inactive. Let's show that a, L, (IC^t), (DC^t), (α_j^c), (R_j), (r_{jt}) satisfy all the constraints of Lemma 5.20, restated below for convenience.

Lemma 5.20. *Let $\rho = 2$ and $\Gamma = 3 + \ln(2)$. Let $\hat{f}$ be any non-negative real number. Let $\varepsilon \geq 0$ be any sufficiently small non-negative real number. Let $([s] \cup \{i\}, d)$ be a Euclidean metric. Let $w : [s] \to \mathbb{R}_{>0}$ be any non-negative weight function. Let $a \in [s]$ be any positive integer such that $a \leq s$. Let $(IC^t)_{t \in [s]}$ and $(DC^t)_{t \in [s]}$ be two families of sets satisfying:*

(a) For each $t \in [s]$, $DC^t \cap IC^t = \emptyset$ and $DC^t \cup IC^t = [t-1]$.

(b) For each $t \in [s-1]$, $DC^t \subseteq DC^{t+1}$.

(c) $[a+1 : s-1] \subseteq IC^s$.

Suppose that the variables $(\alpha_j^*)_{j \in [s]}, (R_j)_{j \in [s]}, (r_{jt})_{t \in [s], j \in DC^t}$ satisfy the following constraints

(1) For any $t \in [s], j \in DC^t$, we have $r_{jt} \leq R_j$.

(2) For each $t \in [a], j \in DC^t$ and $\ell \in IC^t \cup \{t\}$, we have $\alpha_\ell^* \leq (r_{jt} + d(j, \ell))^2$.

(3) For each $t \in [a] \cup \{s\}$, we have

$$\sum_{\ell \in [t:s]} w(\ell) \alpha_t^* + \sum_{\ell \in IC^t} w(\ell) \alpha_\ell^* + \sum_{j \in DC^t} w(j) r_{jt}^2 \leq \hat{f} + \sum_{\ell \in IC^t \cup [t:s]} \rho w(\ell) d^2(\ell, i) + \sum_{j \in DC^t} w(j) d^2(j, i).$$

(4) There is a set $L \subseteq [a] \cap IC^s$ such that:

- *$\sum_{\ell \in L} w(\ell) \leq \varepsilon \cdot \sum_{\ell \in [a+1:s]} w(\ell)$;*
- *for any $j \in DC^s, \ell \in (IC^s \cup \{s\}) \setminus L$, we have $\alpha_\ell^* \leq (1 + \varepsilon) \cdot (r_{js} + d(j, \ell))^2$;*
- *for any $j \in DC^s, \ell \in L$ and $\ell' \in [a+1 : s]$, we have $\alpha_\ell^* \leq (r_{js} + d(j, \ell'))^2$.*

Then, the following holds:

$$\sum_{j \in [s]} w(j) \cdot (\alpha_j^* - (\rho - 1) R_j^2) \leq \hat{f} + (\Gamma + 10\varepsilon) \cdot \sum_{j \in [s]} w(j) \cdot d^2(j, i) \ . \tag{20}$$

Just like for the other cases, constraints (a), (b), (c), (1), (2) and (3) hold. Remains to prove that constraint (4) is satisfied. Because $L \subseteq IC^F$, $IC^F \subseteq [a]$ and $IC^F \subseteq IC^s$, we have $L \subseteq [a] \cap IC^s$. Then, by proving the following claim, we establish constraint (4). Note that because $IC^F \setminus L \cup [a+1 : s] = IC^s \setminus L$, the second and third bullet of the following claim together implies the second bullet of constraint (4). The fourth bullet of the following claim implies the third bullet of constraint (4).

Claim 7.12. *The following hold:*

- $\sum_{\ell \in L} w(\ell) \le \varepsilon \cdot \sum_{\ell \in [a+1:s]} w(\ell)$;
- *for any* $j \in DC^s, \ell \in IC^F \setminus L$, *we have* $\alpha^c_\ell \le (1 + 20\sqrt{\varepsilon}) \cdot (r_{js} + d(j,\ell))^2$;
- *for any* $j \in DC^s$ *and* $\ell \in [a+1:s]$, *we have* $\alpha^c_\ell \le (r_{js} + d(j,\ell))^2$;
- *for any* $j \in DC^s, \ell \in L$ *and* $\ell' \in [a+1:s]$, *we have* $\alpha^c_\ell \le (r_{js} + d(j,\ell'))^2$.

Proof. We prove the four bullets of the claim one by one.

First bullet. By the definition of Case 2(b), the weight bound is our direct assumption. Substituting the dummy variable ℓ for j yields the first bullet immediately.

Second bullet. Consider any $j \in DC^s$ and $\ell \in IC^F \setminus L$. Let θ be the active dual parameter right before i_0 is opened. Because $\ell \notin L$, its distance to i satisfies $d(\ell, i) \le \sqrt{\varepsilon\theta}$. Because i_0 freezes i, we know $d(i, i_0) \le 2\sqrt{\varepsilon\theta}$. By the metric triangle inequality, $d(\ell, i_0) \le 3\sqrt{\varepsilon\theta}$.

To bridge the distance to the open facilities S^s, we instantiate an active client $\ell' \in [a+1:s] \cap B(i_0, \sqrt{\varepsilon\theta})$, which is guaranteed to exist by the Subcase 2(b) definition. Because ℓ' is active, we strictly know $\theta \le (1-\delta)d^2(\ell', S^s)$, which implies $d(\ell', S^s) \ge \sqrt{\theta}$.

Applying the triangle inequality across j, ℓ, ℓ', and S^s yields the core distance relationship:

$$d(\ell, S^s) \ge d(\ell', S^s) - d(\ell, \ell') \ge \sqrt{\theta} - 4\sqrt{\varepsilon\theta} \ .$$

By applying the triangle inequality to j, ℓ, and S^s and scaling by $\sqrt{1-\delta}$, we directly obtain (because $\delta = \varepsilon/2$):

$$(r_{js} + d(j,\ell))^2 \ge \theta \left(\sqrt{1-\delta} - 4\sqrt{\varepsilon}\right)^2 \ge \theta \left(1 - \varepsilon/2 - 4\sqrt{\varepsilon}\right)^2 > \theta(1 - 9\sqrt{\varepsilon}) \ .$$

Squaring both sides and applying the Taylor expansion $(1 - 9\sqrt{\varepsilon})^{-1} \le 1 + 10\sqrt{\varepsilon}$ (valid for our sufficiently small ε) provides the exact upper bound for θ:

$$\theta \le (1 + 10\sqrt{\varepsilon})(r_{js} + d(j,\ell))^2 \ .$$

Because $\ell \in IC^F$, it is indirectly connected right before i_0 opens, meaning its modified dual value is strictly bounded by the active parameter: $\alpha^c_\ell \le \theta$. Substituting our geometric bound yields the final inequality:

$$\alpha^c_\ell \le (1 + 20\sqrt{\varepsilon})(r_{js} + d(j,\ell))^2 \ .$$

Third bullet. Consider any $j \in DC^s$ and $\ell \in [a+1:s]$. Because ℓ is active right before i_0 is opened, its maximum possible modified dual value is dictated by the second bullet of the openability condition (Definition 7.1), which strictly enforces $\tau_\ell \le (1-\delta)d^2(\ell, S^s)$. Thus, we bypass the parameter θ entirely:

$$\alpha^c_\ell \le \tau_\ell \le (1-\delta)d^2(\ell, S^s) \ .$$

Applying the triangle inequality to j, ℓ, and S^s, we have $d(\ell, S^s) \le d(j, S^s) + d(j, \ell)$. Multiplying by $\sqrt{1-\delta}$ and substituting $r_{js} = \sqrt{1-\delta}d(j, S^s)$ yields:

$$\sqrt{1-\delta}d(\ell, S^s) \le r_{js} + \sqrt{1-\delta}d(j,\ell) \le r_{js} + d(j,\ell) \ .$$

Squaring both sides and substituting into our openability bound gives exactly:

$$\alpha^c_\ell \le (1-\delta)d^2(\ell, S^s) \le (r_{js} + d(j,\ell))^2 \ ,$$

completing the proof of the third bullet.

Fourth bullet. Consider any $j \in DC^s$, $\ell \in L$, and $\ell' \in [a+1:s]$. Because ℓ' is active right before i_0 is opened, we strictly have $\theta \le (1-\delta)d^2(\ell', S^s)$.

Applying the triangle inequality to j, ℓ', and S^s, we have $d(\ell', S^s) \le d(j, S^s) + d(j, \ell')$. Multiplying by $\sqrt{1-\delta}$ yields:

$$\sqrt{1-\delta}d(\ell', S^s) \le r_{js} + \sqrt{1-\delta}d(j, \ell') \le r_{js} + d(j, \ell') \ .$$

Squaring both sides provides the geometric upper bound for θ:

$$\theta \le (1-\delta)d^2(\ell', S^s) \le (r_{js} + d(j, \ell'))^2 \ .$$

Finally, because $\ell \in L \subseteq IC^F$, it is strictly indirectly connected. According to the algorithm, its dual value is never increased during Stage 1 updates, strictly ensuring $\alpha_\ell^c \le \theta$. Combining this with our geometric upper bound gives the exact final relation:

$$\alpha_\ell^c \le \theta \le (r_{js} + d(j, \ell'))^2 \ ,$$

completing the proof of the fourth bullet. □

Thus, we can apply Lemma 5.20. We get:

$$\sum_{j \in [s]} w(j) \cdot (\alpha_j^c - (\rho-1)R_j^2) \le \hat{f} + (\Gamma + 200\sqrt{\varepsilon}) \cdot \sum_{j \in [s]} w(j) \cdot d^2(j, i).$$

This completes the proof.

8 Approximation with $O(\log n/\varepsilon^3)$ Extra Centers

In this section, we present bi-criteria approximations with $O(\log n/\varepsilon^3)$ extra centers, as claimed in Theorem 1.3 (restated in Theorem 8.7). For simplicity, we will assume that $\rho \in [2, 3]$ in the rest of this section.

8.1 Pre-processing

For this section, we will consider the case where the distances between clients and facilities are lower bounded by 1, thanks to the following variant of Lemma 3.3:

Lemma 8.1. *For any given $\varepsilon > 0$ and $\alpha > 1$, given a polynomial-time α-approximation algorithm for Euclidean k-Means on instances with distances between 1 and $2n^3/\varepsilon^3 + 1$, there exists a polynomial-time $\alpha(1+O(\varepsilon))$-approximation algorithm for Euclidean k-Means on general instances.*

Proof. Given Lemma 3.3, it suffices to prove the following argument:

> For any given $\varepsilon > 0$ and $\alpha > 1$, given a polynomial-time α-approximation algorithm for Euclidean k-Means on instances with distances between 1 and $2n^3/\varepsilon^3 + 1$, there exists a polynomial-time $\alpha(1 + O(\varepsilon))$-approximation algorithm for Euclidean k-Means on instances with non-zero distances between 1 and n^2/ε^2.

Suppose that we have a Euclidean k-Means instance with non-zero distances between 1 and n^2/ε^2 (namely, the old instance), in which $x : D \cup F \to \mathbb{R}^{\mathsf{dim}}$ denotes the coordinate of each point. Now, we index $D \cup F$ by $1, 2, \ldots, |D| + |F|$ arbitrarily. We construct the following instance (namely, the new instance) by assigning each point at $x' : D \sqcup F \to \mathbb{R}^{\mathsf{dim}+|D|+|F|}$, where $x'(i) = (\frac{2n}{\varepsilon} \cdot x(i), e_i)$ for each $i \in D \sqcup F$, and $e_i \in \mathbb{R}^{|D|+|F|}$ denotes the indicator vector, which has a value of 1 on coordinate i and a value of 0 on all other coordinates. Note that we will separate two copies for $i \in D \cap F$ in the new instance. Let d and d' denote the Euclidean metric induced by the old and new instances, respectively. Consider any $i \in F, j \in D$. We have

$$d'^2(i,j) = \frac{4n^2}{\varepsilon^2}\|x(i) - x(j)\|^2 + \|e_i - e_j\|^2 = \frac{4n^2}{\varepsilon^2}d^2(i,j) + 2 \ . \tag{38}$$

It is easy to see that $d'(i,j) \in [1, \frac{2n}{\varepsilon}d(i,j) + 1] \subseteq [1, 2n^3/\varepsilon^3 + 1]$.

Now, we assume that we have an α-approximation on the new instance. We first run a constant-factor approximation k-Center on the old instance and decide if the optimal value for the old instance OPT on k-Means is 0. If it's not, it is guaranteed that OPT is at least 1. According to Equation (38), the optimal value of the new instance OPT' is $4n^2/\varepsilon^2 \cdot OPT + 2n \geq 4n^2/\varepsilon^2 + 2n$. Then we run the α-approximation on the new instance, getting a solution with value ALG' on the new instance such that $ALG' \leq \alpha \cdot OPT'$. We obtain the solution ALG on the old instance with the same set of open facilities and the same connections for the clients. Again, because of Equation (38), we have $ALG' = 4n^2/\varepsilon^2 \cdot ALG + 2n$. This implies, on the old instance, we have

$$\frac{ALG}{OPT} = \frac{(\varepsilon^2/4n^2) \cdot (ALG' - 2n)}{(\varepsilon^2/4n^2) \cdot (OPT' - 2n)} = \frac{ALG' - 2n}{OPT' - 2n} = \frac{ALG' - 2n}{OPT'} \cdot \frac{OPT'}{OPT' - 2n} \leq \alpha(1 + \varepsilon^2/(4n^2)) \ .$$

This completes the proof of this lemma. □

We shall remark here that under this assumption, our log-adaptive Algorithm 2 still runs in $O(\log n/\varepsilon^3)$ phases.

8.2 Walking Between Two Solutions: Setup

This subsection is an adaptation of Subsection 5.2 of [CCG+26] to our new bidding strategy.

Given the description of the log-adaptive algorithm, we present our final bicriteria approximation MergeSolutions. It results in a solution that opens $k + O(\log n)$ centers by maintaining two partial solutions that lead to opening at least k and at most k centers respectively, and *gradually merging them* into a single solution that opens exactly $k + O(\log n)$ facilities, exploiting the $O(\log n)$ adaptivity of the algorithm.

Before detailing the merging procedure MergeSolutions, it is helpful to observe that we may generalize the analysis to make it more "robust". We introduce this more robust analysis in this subsection, where we also define the notation that will be used later to describe MergeSolutions.

Starting from the two solutions obtained in the standard way whose facility costs differ by at most $\eta := 2^{-n}$, the two solutions we maintain will have an exponentially small difference η in one of their parameters. (This parameter will be either the facility cost f, some distance related to a *free facility* that will be introduced later, or some probability that a client is in IC or DC.) For this reason, when we merge the two solutions, it may occur that some facilities in one solution are almost completely paid for, rather than fully paid for, or that the dual

constraint of some facilities is slightly violated. To address this, we generalize the definition of "openable" to "η-openable", where the only difference is an $O(n\eta)$ slack in the third and fourth conditions.

Note that, in the algorithm, each inactive client j is always either directly connected to some facility i^* or indirectly connected to some facility i^*, where i^* is given by $i^* = \arg\min_{i \in S} d^2(j,i)$. In the rest of this section, we will maintain this invariant, but allow some flexibility to its connection type and α-value. Specifically, we allow each client $j \in D$ to have a set of copies D_j, with each $\tilde{j} \in D_j$ defined by (i) the connection type $ct(\tilde{j})$, i.e., whether it is in A, IC or DC; (ii) the α-value $\tilde{\alpha}(\tilde{j})$; (iii) the probability mass $\mu(\tilde{j})$. The probability mass should satisfy $\sum_{\tilde{j} \in D_j} \mu(\tilde{j}) = 1$. In particular, for each client j, D_j has two possibilities:

- D_j has only one copy that is of type A and with probability mass 1;
- D_j has some copies of type DC and at most one copy of type IC.

In short, each D_j contains at most one copy that is of type IC or A. We will say j is active if the first bullet holds, and inactive if the second bullet holds. In this way, we define the α^I-value of a client as the α-value of its active copy or the indirectly connected copy. If the client has no such copy, we say its α^I-value is 0 (as this value will not be used). We remark this α^I-value is different from the α-values of the copies, and will only be used when we merge two solutions (Subsection 8.3). Moreover, we will treat each copy $\tilde{j}$ as a normal weighted client located at the same point as j in the metric space. When we compute the total bid, we average out over their different copies.

Further, we consider $\mathcal{D}$ as *the collection of (succinct) copies* by dropping all the α-values of directly connected copies from $\cup_{j \in D} D_j$. This collection will be used to define the algorithm that merges two solutions. In this way, we can still deduce α^I values for each client and this set can be independent of the α-values of directly connected copies, which are not necessary in defining the algorithm.

Equivalently, for each inactive client j, we consider its connection as a discrete probability distribution over (i) the connection type ct_j, i.e., whether it is in IC or DC; (ii) the α-value of the client α_j. In particular, in the support of this distribution, there can be only one possible α-value for indirect connects, i.e., there exists at most one value α^I such that $\Pr[(ct_j, \alpha_j) = (IC, \alpha^I)] > 0$. When we compute the total bid, we average out their contributions. Adapting to our new dual fitting approach, the expected contribution of client j to facility i now becomes:

$$\begin{aligned}&\mathbb{E}\left[(1-\delta)[d^2(j,S) - d^2(j,i)]^+ \mid ct_j = DC\right] \cdot \Pr[ct_j = DC] \\ &\qquad + \mathbb{E}\left[[\alpha_j - \rho(1-\delta)d^2(j,i)]^+ \mid ct_j = IC\right] \cdot \Pr[ct_j = IC],\end{aligned}$$

where the probability and the expectations are all taken over the discrete distribution. Letting $\pi(j) = \Pr[ct_j = DC]$ and letting α_j^I be the only possible α-value for indirect connects of j, we can rewrite the above as:

$$\pi(j) \cdot (1-\delta)[d^2(j,S) - d^2(j,i)]^+ + (1-\pi(j)) \cdot [\alpha_j^I - \rho(1-\delta)d^2(j,i)]^+$$

For each active client j, the connection distribution is a distribution with one element in the support having type $ct_j = A$. Therefore, we can define $\pi(j) = 0$ and let α_j^I be the α-value of j. In the rest of this section, we will use both points of view interchangeably.

Given the time $\theta = (1+\varepsilon^3)^{p-1}$, the sequence $\mathcal{H} = (H_1, \ldots, H_p)$ of the sequences of facilities opened by each stage 1 of the algorithm that was executed so far, we easily can obtain the set

of open facilities S. Further given the collection of copies $\mathcal{D}$ for the clients at the beginning of the phase, we can deduce the α-value for each indirectly connected or active copy. Then, we can use the following process RetrieveNewCopies, which is the third step of stage 1 of our modified log-adaptive algorithm, to obtain the current collection of copies:

RetrieveNewCopies $(\mathcal{D}, H_p)$
For each copy $j \in \mathcal{D}$ that is active or indirectly connected:

- Let α^I be its α-value and j be the client it corresponds to.
- If $\alpha^I \geq \rho(1-\delta)d^2(j, H_p)$, we modify the copy to be directly connected and drop its α-value.
- Else if $\alpha^I \geq (1-\delta)d^2(j, H_p)$, we modify the copy to be indirectly connected and keep its α-value unchanged.

Given the output of the procedure, we can deduce all current copies for each client, and their corresponding connection types, probability mass and α-values. Based on these variables, we have a new definition of η-openability tailored for our new dual fitting approach.

Definition 8.2 (η-openable)**.** *Consider the time $\theta = (1+\varepsilon^3)^{p-1}$. Let $\mathcal{H} = (H_1, \ldots, H_{p-1}, H_p)$ be the sequence of sequences of opened facilities so far and let $\mathcal{D}$ be the collection of copies for all clients at the beginning of the phase p. Then sets S, A, IC, DC can be deduced from $\mathcal{H}$ and* RetrieveNewCopies $(\mathcal{D}, H_p)$ *defined as above.*

For each copy j, we can also deduce its probability mass $\mu(j)$, and its α-value α_j if $j \in IC \cup A$. We say that a facility $i \in F$ is η-openable *(with respect to θ, $\mathcal{H}$ and $\mathcal{D}$) if there are $(\tau_j)_{j \in A}$ that satisfy the following conditions.*

- *Only nearby clients are increased:*

$$\tau_j = \alpha_j \text{ for every } j \in A \setminus B(i, \sqrt{\varepsilon\theta}).$$

- *Nearby clients are only increased slightly:*

$$\alpha_j \leq \tau_j \leq \min\{(1-\delta)d^2(j,S), (1+\varepsilon^3)\theta\} \text{ for every } j \in A \cap B(i, \sqrt{\varepsilon\theta}).$$

- *Facility i is almost paid for:*

$$(1-\delta)\sum_{j \in DC} \mu(j) \cdot [d^2(j,S) - d^2(j,i)]^+ + \sum_{j \in IC} \mu(j) \cdot [\alpha_j - \rho(1-\delta)d^2(j,i)]^+ + \sum_{j \in A} [\tau_j - \rho(1-\delta)d^2(j,i)]^+ \geq \hat{f} - 8n\eta.$$

- *Dual feasibility almost holds: for every facility i_0,*

$$\sum_{j \in DC} \mu(j) \cdot [(1-\delta)d^2(j,S) - d^2(j,i_0)]^+ + \sum_{j \in IC} \mu(j) \cdot [\alpha_j - \rho d^2(j,i_0)]^+ + \sum_{j \in A} [\tau_j - \rho d^2(j,i_0)]^+ \leq \hat{f} + 8n\eta.$$

Remark 8.3. *Equivalently, the third bullet can be rewritten in the point of view of the clients (instead of the copies): Suppose A is the set of active clients and α_j^I is the α^I-value for client j, and $\pi(j)$ is the probability that $j \notin A$ is directly connected. Then, the four bullets can be rewritten as*

- *Only nearby clients are increased:*

$$\tau_j = \alpha_j^I \text{ for every } j \in A \setminus B(i, \sqrt{\varepsilon\theta}).$$

- *Nearby clients are only increased slightly:*

$$\alpha_j^I \le \tau_j \le \min\{(1-\delta)d^2(j,S), (1+\varepsilon^3)\theta\} \text{ for every } j \in A \cap B(i, \sqrt{\varepsilon\theta}).$$

- *Facility i is almost paid for:*

$$\sum_{j \notin A} \left(\pi(j) \cdot (1-\delta) \cdot [d^2(j,S) - d^2(j,i)]^+ + (1-\pi(j)) \cdot [\alpha_j^I - \rho(1-\delta)d^2(j,i)]^+ \right) + \sum_{j \in A} [\tau_j - \rho(1-\delta)d^2(j,i)]^+ \ge \hat{f} - 8n\eta.$$

- *Dual feasibility almost holds: for every facility i_0,*

$$\sum_{j \notin A} \left(\pi(j) \cdot \left[(1-\delta)d^2(j,S) - d^2(j,i_0)\right]^+ + (1-\pi(j)) \cdot \left[\alpha_j^I - \rho d^2(j,i_0)\right]^+ \right) + \sum_{j \in A} [\tau_j - \rho d^2(j,i_0)]^+ \le \hat{f} + 8n\eta.$$

Similar as [CGL+25; CCG+25; CCG+26], we introduce the concept of *free facilities*. When we merge the two solutions, we introduce free facilities, whose opening costs are not necessarily paid. We will introduce at most $O(1)$ such free facilities in each phase. However, we do not restrict the number of free copies a regular facility may have. Let $S_{\mathsf{free}} \subseteq \mathcal{F}$ denote the multiset of free facilities that have been opened.

Moreover, we augment the metric space for the free facilities. For each free copy $\tilde{i} \in S_{\mathsf{free}}$ of a regular facility $i \in \mathcal{F}$, we define a parameter $u(\tilde{i}) \ge 0$, which acts as the squared distance overhead of the free copy. Therefore, for any point x in the metric space (either a facility or a client), the distance squared is given by $d^2(\tilde{i}, j) = u(\tilde{i}) + d^2(i, j)$.

We further introduce the following terminology: as mentioned, we refer to the facilities in S_{free} as *free facilities*, the facilities in $\mathcal{F}$ and $S_{\mathsf{reg}} = S - S_{\mathsf{free}}$ as *regular* facilities, and when we simply mention "facilities", we are referring to their union.

Note that if d is Euclidean, we can have one additional dimension, which equals $\sqrt{u(\tilde{i})}$ for any free facility $\tilde{i}$ and equals 0 for any regular facility or client, to maintain the Euclidean property.

Finally, we will use the letters i and h to denote facilities. Specifically, we use $i \in \mathcal{F}$ to denote a regular facility and use $\tilde{i}$ to denote a free copy of i. When referring to a facility that can be either free or regular, we use the letter h.

We now introduce the definition of valid sequences, which aims to capture the sequence of facilities that are opened during the while-loop of stage 1 of the log-adaptive algorithm.

Let us say $\mathcal{H}' = (H'_1, \ldots, H'_p)$ is a super-sequence of $\mathcal{H} = (H_1, \ldots, H_p)$ and denote it $\mathcal{H}' \supseteq \mathcal{H}$ if they have a common prefix of length $p-1$, i.e., for all $i < p$, $H'_i = H_i$, and H'_p is a super-sequence of H_p. Given two sequences $\mathcal{H}_1 = (H_1, \ldots, H_r)$ and $\mathcal{H}_2 = (H_{r+1}, \ldots, H_l)$, we define $\mathcal{H}_1 \oplus \mathcal{H}_2 = (H_1, \ldots, H_l)$ as the concatenation of the two sequences.

Definition 8.4 (η-valid sequence)**.** *Consider the time θ. Let $\mathcal{H}$ be the sequence of sequences of opened facilities and $\mathcal{D}$ be the collection of copies at the beginning of the phase. A sequence $\langle h_1, \ldots, h_\ell \rangle$ of facilities is η-valid (with respect to $\theta, \mathcal{D}$ and $\mathcal{H}$), if the following conditions hold.*

- *Each h_t is either free or η-openable with respect to the time θ, a super-sequence $\mathcal{H}' \supseteq \mathcal{H} \oplus (\langle h_1, \ldots, h_{t-1} \rangle)$ of the sequence of sequences of opened facilities, where $\oplus$ is just the concatenation, and the collection $\mathcal{D}$.*
- *The sequence is maximal in terms of (0-)openability, i.e., there is no (0-)openable facility with respect to the time θ, $\mathcal{H} \oplus (\langle h_1, \ldots, h_\ell \rangle)$, and the collection $\mathcal{D}$.*

We consider the following procedure CompleteCopies to return the collection of copies after opening a sequence of facilities H_p during the first stage of the phase. This procedure is identical to executing stage 2 of the log-adaptive algorithm on the collection of copies.

CompleteCopies $(\theta, \mathcal{D}, \mathcal{H} = (H_1, \ldots, H_p))$

Set $S = \cup_{r \in [p]} H_r$ and $\mathcal{D}' =$ RetrieveNewCopies$(\mathcal{D}, H_p)$. For each copy $j \in \mathcal{D}'$:

- If j is of type IC, set $\alpha_j \leftarrow (1-\delta) d^2(j, S)$.
- If j is of type A, set $\alpha_j \leftarrow \min\{(1+\varepsilon^3)\theta, (1-\delta) d^2(j, S)\}$, and change its connection type to IC when $\alpha_j = (1-\delta) d^2(j, S)$.

Definition 8.5 (Valid collection of copies)**.** *Consider the time $\theta = (1+\varepsilon^3)^{p-1}$. Let $\mathcal{H} = (H_1, \ldots, H_p)$ be the sequence of sequences of opened facilities and $\mathcal{D}$ be the collection of copies after the first $p-1$ phases. A collection of copies $\mathcal{D}'$ is valid (with respect to $\theta, \mathcal{H}, \mathcal{D}$), if $\mathcal{D}'$ can be obtained by the following procedure.*

> *Suppose $\mathcal{D}'' =$ CompleteCopies$(\theta, \mathcal{D}, \mathcal{H})$. Then, we can obtain $\mathcal{D}'$ by the following process: for each copy j that is indirectly connected or active in $\mathcal{D}$ but is directly connected in $\mathcal{D}''$, if $\rho(1-\delta) d^2(j, S) \le \alpha_j \le \rho(1-\delta) d^2(j, S) + 4\eta$ (where α_j is the α-value of j in $\mathcal{D}$), then it can be split into a directly connected copy with an arbitrarily smaller probability mass, while the remaining probability mass is given to an indirectly connected copy with α-value $\alpha'_j = (1-\delta) d^2(j, S)$.*

In particular, if $\mathcal{D}' = \mathcal{D}''$, we say that $\mathcal{D}'$ is perfectly valid.

We remark that

- The log-adaptive algorithm generates 0-valid sequences in the first stage of each phase. Specifically, it generates sequences $\mathcal{H} = (H_1, H_2, \ldots, H_L)$ and a history of the collections of copies $\mathcal{DH} = (\mathcal{D}_0, \mathcal{D}_1, \ldots, \mathcal{D}_L)$, where each H_p is 0-valid (with respect to $\theta = (1+\varepsilon^3)^{p-1}$, $\mathcal{D}_{p-1}$ and $\mathcal{H}$) and contains only regular facilities.

- The log-adaptive algorithm only maintains a single copy for each inactive client, i.e., each inactive client j is either directly connected to some facility i with probability 1 or indirectly connected to some facility i with probability 1. Therefore, each collection of copies $\mathcal{D}_p$ maintained by the log-adaptive algorithm is perfectly valid (with respect to $\theta = (1+\varepsilon^3)^{p-1}$, $\mathcal{H}$ and $\mathcal{D}_{p-1}$).
- The set S of opened facilities is such that $A = \emptyset$ at the final time θ.

For a general sequence of sequences $\mathcal{H} = (H_1, H_2, \ldots, H_L)$ and a history of collection of copies $\mathcal{DH} = (\mathcal{D}_0, \mathcal{D}_1, \ldots, \mathcal{D}_L)$, let us say they consist of η-valid sequences and valid collections of copies if, for every $p \in [L]$,

- H_p is an η-valid sequence with respect to $\theta = (1+\varepsilon^3)^{p-1}$, $\mathcal{H}$, $\mathcal{D}_{p-1}$; and
- $\mathcal{D}_p$ is a valid collection of copies with respect to $\theta = (1+\varepsilon^3)^{p-1}$, $\mathcal{H}$, $\mathcal{D}_{p-1}$.

It is easy to see from the definitions that the following claim holds.

Claim 8.6. *Suppose that $\mathcal{H}, \mathcal{DH}$ consist of η-valid sequences and valid collections of copies. For each $p \in [L]$, letting $S = \cup_{r\in[p]} H_r$, then we have*

- *If $j \in \mathcal{D}_p$ is indirectly connected, then $\alpha_j = (1-\delta)d^2(j,S)$;*
- *If $j \in \mathcal{D}_p$ is active, then $\alpha_j = (1+\varepsilon^3)^p$ and $(1+\varepsilon^3)^p < (1-\delta)d^2(j,S)$.*

Furthermore, we say that $\mathcal{H}, \mathcal{DH}$ form a *solution* with open facilities $S = \cup_{r\in[L]} H_r$ if there are no active copies in $\mathcal{D}_L$. Otherwise, we say that $\mathcal{H}, \mathcal{DH}$ form a partial solution.

The results in Section 7, namely Lemmas 7.3 and 7.4, can thus be stated as: if $\mathcal{H}, \mathcal{DH}$ is a solution of 0-valid sequences consisting of regular facilities, then the set S of opened facilities satisfies

$$\sum_{j\in D} w(j)\cdot d^2(j,S) \le \frac{\Gamma}{1-\delta}\cdot(\mathrm{opt}_{\mathrm{LP}}(f) - f\cdot|S|)\ .$$

We now allow an additive $8n\eta$ slack in the third bullet (facility paying) and fourth bullet (dual feasibility) of the openability (Definition 8.2). In addition, in the second stage of each phase, we allow a directly connected copy j that satisfies $\rho(1-\delta)d^2(j,S) \le \alpha_j < \rho(1-\delta)d^2(j,S) + (1+\rho)\eta$ to split into two copies with arbitrary probability mass: one is identical, while the other is indirectly connected with α-value $\alpha_j = (1-\delta)d^2(j,S)$. All these modifications lead to the following generalization.

Theorem 8.7. *Consider a solution $\mathcal{H}, \mathcal{DH}$ of η-valid sequences and valid collections of copies. Let S be the opened facilities. Further, let S_{free} and $S_{\mathsf{reg}} = S - S_{\mathsf{free}}$ be the free copies and regular facilities, respectively. Then, we have*

$$\sum_{j\in D} d^2(j,S) \le \frac{\Gamma + 201\sqrt{\varepsilon} + 20n\eta}{1-\delta}\cdot(opt_{LP}(f) - |S_{\mathsf{reg}}|\cdot f)\ .$$

where

- *if d is a (general) metric, we choose $\rho = 2.5, \Gamma = 4.9$; or*
- *if d is a Euclidean metric, we choose $\rho = 2, \Gamma = 3 + \ln(2)$.*

In the next subsection, we will merge two solutions into one with $|S_{\mathsf{reg}}| = k$ and $|S_{\mathsf{free}}| = O(\log n/\varepsilon^3)$. To our help, we will use two basic routines defined as follows:

- CompleteSolution$_{f,u}(\mathcal{H}, \mathcal{DH})$ takes a partial solution $\mathcal{H}, \mathcal{DH}$ as input and returns a solution by running the while-loop of the log-adaptive algorithm using f, u and the existing copies induced by the last collection in $\mathcal{DH}$.

  In particular, we also consider a slightly different input for this routine. If $\mathcal{H} = (H_1, \ldots, H_p)$ and $\mathcal{DH} = (\mathcal{D}_0, \ldots, \mathcal{D}_{p-1})$, where the first $p-1$ sequences of $\mathcal{H}$ form a partial solution with $\mathcal{DH}$, and H_p is η-valid. Then, we will first generate $\mathcal{D}_p$ by CompleteCopies $(\theta = (1+\varepsilon^3)^{p-1}, \mathcal{D}_{p-1}, (H_1, \ldots, H_p))$, append it to the end of $\mathcal{DH}$ (so that the resulting $\mathcal{H}, \mathcal{DH}$ form a partial solution), and then start from phase $p+1$ of the log-adaptive algorithm.

- CompleteSequence$_{f,u}(\mathcal{H} = (H_1, \ldots, H_{p-1}), H_p = \langle h_1, \ldots, h_\ell \rangle, \mathcal{DH} = (\mathcal{D}_0, \ldots, \mathcal{D}_{p-1}))$ where $\mathcal{H}, \mathcal{DH}$ form a partial solution, and H_p is an η-valid sequence except that it may not be maximal. CompleteSequence then runs the while-loop of the log-adaptive algorithm starting with H_p and the existing copies induced by the last collection $\mathcal{DH}$ to return a maximal sequence of H_p (with respect to f, u).

We note that all facilities added by CompleteSolution and CompleteSequence are 0-openable and regular. We also note that, given $\mathcal{D}_{p-1}$ and $\mathcal{H}$, CompleteSolution generates $\mathcal{D}_p$ by CompleteCopies $(\theta = (1+\varepsilon^3)^{p-1}, \mathcal{D}_{p-1}, (H_1, \ldots, H_p))$. Additionally, for intuition, observe that, with the above notation, the set of facilities opened by CompleteSolution$_{f,u}(\emptyset, (\mathcal{D}_0))$ is equivalent to that opened by the log-adaptive algorithm, where $\mathcal{D}_0$ denotes the collection of copies in which each client has only one copy with $(ct = A, \mu = 1, \alpha = 1)$. (There is no free facility here.) To simplify notation, we only use CompleteSequence when the partial solution $\mathcal{H}$ and $\mathcal{DH}$ are clear from the context, so we simply write CompleteSequence$_{(f,u)}(\langle h_1, \ldots, h_\ell \rangle)$. Furthermore, we omit (f, u) when the parameters are clear from the context.

8.3 MergeSolutions

This subsection is almost identical to Subsection 5.3 of [CCG+26]. The main difference is the proof of Claim 8.8 in Subsection 8.3.1.

We now present our procedure MergeSolutions that *walks* between two solutions. We will maintain two solutions $(\mathcal{H}, \mathcal{DH}, f, u)$ and $(\mathcal{H}', \mathcal{DH}', f', u')$ whose numbers of open regular facilities *sandwich* k. (Formally, let us say a and b sandwich k if $a < k < b$ or $b < k < a$.) We will gradually make them closer until one of them opens exactly k regular facilities. We start from the two initial solutions obtained using the standard method. Note that the minimum nonzero distance is 1 and the maximum squared distance is $M = \mathsf{poly}(n)$, which implies that $\mathrm{opt} \le nM$.

Initialization. We binary search on f in the interval $[1/n^2, 4nM]$ and run the routine CompleteSolution$_{f,u}(\emptyset, \langle \mathcal{D}_0 \rangle)$ (with zero u). When $f = 1/n^2$, every client $j \in D$ satisfies $d(j, S) = 0$ if $d(j, \mathcal{F}) = 0$ and $d^2(j, S) \le d^2(j, \mathcal{F}) + 2/n^2$ otherwise, as each client can open its closest facility by itself when $\alpha_j = d^2(j, \mathcal{F}) + 2/n^2$. If this solution opens at most k centers, then it is already a $(1 + 2/n)$-approximation. On the other hand, when $f = 4nM$, the algorithm will open exactly one facility, since the first facility is open when $\theta \ge 2M$, so every client becomes immediately inactive when it is open. Therefore, the binary search yields two solutions $(\mathcal{H}, \mathcal{DH}, f, u)$ and $(\mathcal{H}', \mathcal{DH}', f', u)$ where $(\mathcal{H}, \mathcal{DH}, f, u)$ opens less than k facilities and $(\mathcal{H}', \mathcal{DH}', f', u)$ opens more than k facilities where $f \le f' \le f + \eta$.

General Step. MergeSolutions works on phase $p = 1, 2, \ldots$ iteratively, and ensures that the two solutions are identical up to the p-th phase. Formally, MergeSolutions takes two solutions $(\mathcal{H}, \mathcal{DH}, f, u)$ and $(\mathcal{H}', \mathcal{DH}', f', u')$ as well as the current phase p as input. The following promises on them will be satisfied whenever we call MergeSolutions.

(i) Both consist of η-valid sequences.

(ii) The numbers of open regular facilities by the two solutions sandwich k.

(iii) At most one parameter, which we call the *difference parameter*, differs by at most η between the two solutions, i.e., if $f \neq f'$ then $|f - f'| \leq \eta$ and $u(\tilde{i}) = u'(\tilde{i})$ for every $\tilde{i} \in S_{\mathsf{free}}$; otherwise if $f = f'$ there is at most one free facility $\tilde{i} \in S_{\mathsf{free}}$ so that $u(\tilde{i}) \neq u'(\tilde{i})$ and $|u(\tilde{i}) - u'(\tilde{i})| \leq \eta$. (We don't consider π, π' defined in the next bullet as *difference parameters*.)

(iv) The probability that each client is directly connected to any facility is almost the same in both solutions, i.e., we have that for every client j, $|\pi(j) - \pi'(j)| \leq \eta/M$, where π, π' are the probabilities of direct connections for the collection of copies $\mathcal{D}_{p-1}$ and $\mathcal{D}'_{p-1}$, respectively.

(v) $\mathcal{H} = (H_1, H_2, \ldots, H_L)$ and $\mathcal{H}' = (H'_1, H'_2, \ldots, H'_{L'})$ have a common prefix of length $p - 1$, i.e., $H_i = H'_i$ for $i = 1, 2, \ldots, p - 1$. Furthermore, the remaining sequences $H_p, \ldots, H_L$ and $H'_p, \ldots, H'_{L'}$ only contain regular facilities and are 0-valid with respect to the parameters f, u and f', u', respectively. (This implies that $\mathcal{H}$ and $\mathcal{H}'$ open the same set of free facilities.)

Given the promises, the goal of MergeSolutions is to

(1) either find a solution of η-valid sequences that opens exactly k regular facilities and $O(\log n)$ free facilities, or

(2) produce two new solutions that satisfy the above promises for phase $p + 1$.

Since the maximum squared distance between any two points is $M \leq \mathsf{poly}(n)$ and $f \leq 4nM$, before θ reaches $6M$, at least one facility will open and every client will become inactive. Therefore, for some threshold $p^* = O(\log n/\varepsilon^3)$, once $\mathcal{H}$ and $\mathcal{H}'$ agree on the first p^* phases, they open exactly the same set of facilities which violates the promise that the numbers of open regular facilities sandwich k. Therefore, case (1) must happen for some $p \leq p^*$, which yields a solution of η-valid sequences that opens exactly k regular facilities and $O(\log n)$ free facilities. Combined with Theorem 8.7, this proves our main bicriteria guarantee. Indeed, we then have a solution consisting of open centers S with $|S| = k + O(\log n/\varepsilon^3)$ such that, by Theorem 8.7,

$$\sum_{j \in D} w(j) \cdot d^2(j, S) \leq \frac{\Gamma}{1 - \delta} \cdot (\mathrm{opt}_{\mathrm{LP}}(f) - (f - \eta n) \cdot k) \leq (\Gamma + O(\varepsilon))\mathrm{opt}\,,$$

where, for the last inequality, we used that $\eta = 2^{-n}$ and $\mathrm{opt} \geq 1$ (since any non-zero distance is at least 1). The main theorem then follows by scaling ε appropriately.

In the remainder of this section, we present MergeSolutions for two solutions and a given phase p. It consists of four subroutines, presented in the subsequent subsections respectively. Each of the first three subroutines achieves (1) or (2) without condition (iv) above (which ends the whole procedure or lets us move on to the fourth subroutine), or sets up the stage for the next subroutine. The third subroutine always achieves (1) or (2) without condition (iv). The fourth subroutine, then achieves (2), for those not achieving (1), by ensuring condition (iv).

8.3.1 Making Parameters Identical

By renaming, we assume that if $f \neq f'$ is the difference parameter then $f' < f$, and if the difference parameter is $u(\tilde{i}) \neq u'(\tilde{i})$ then $u(\tilde{i}) < u'(\tilde{i})$.

Claim 8.8. *$H'_1, H'_2, \ldots, H'_p$ are η-valid sequences with respect to parameters f, u and $\mathcal{D}_0, \mathcal{D}_1, \ldots, \mathcal{D}_{p-1}$, respectively. On the other hand, $\mathcal{D}_1, \ldots, \mathcal{D}_{p-1}$ are valid collections of copies with respect to parameters f, u and $(H'_1, \ldots, H'_{p-1})$.*

Proof. As the first $p-1$ sequences are identical, we have that $H'_1, \ldots, H'_{p-1}$ are η-valid with respect to parameters f, u and $\mathcal{D}_0, \ldots, \mathcal{D}_{p-2}$, respectively. For the same reason, $\mathcal{D}_1, \ldots, \mathcal{D}_{p-1}$ are valid with respect to parameters f, u and $(H'_1, \ldots, H'_{p-1})$. It remains to show that H'_p is η-valid with respect to parameters f, u and $\mathcal{D}_{p-1}$. With an abuse of notation, we will use $\mathcal{H}$ to denote the sequence of sequences $(H'_1, \ldots, H'_{p-1})$ in the rest of the proof.

Now suppose that $|f - f'| \leq \eta$ and $u(\tilde{i}) \neq u'(\tilde{i})$ (if exists). If such $\tilde{i}$ does not exist, we take any $\tilde{i}$. H'_p was 0-valid with respect to $f', u', \mathcal{D}'_{p-1}$. We have that $u'(\tilde{i}) - \eta \leq u(\tilde{i}) \leq u'(\tilde{i})$. Further, $|\tilde{\pi}(j) - \tilde{\pi}'(j)| \leq \eta/M$ for each client j, where $\tilde{\pi}$ and $\tilde{\pi}'$ are the probabilities that j is directly connected in $\mathcal{D}_{p-1}$ and $\mathcal{D}'_{p-1}$, respectively.

Let $H'_p = \langle h_1, \ldots, h_\ell \rangle$ and consider h_t for some $t \in [\ell]$, which is 0-openable with respect to $f', u', \theta = (1+\varepsilon^3)^{p-1}, \mathcal{D}'_{p-1}$ and some super-sequence $\mathcal{H}' \supseteq \mathcal{H} \oplus (\langle h_1, \ldots, h_{t-1} \rangle)$. Let H_{last} be the final sequence in $\mathcal{H}'$. Let S, S' be the set of facilities open in $\mathcal{H}$ and $\mathcal{H}'$, respectively.

Let $d'(\cdot, \cdot)$ denote the pairwise distance defined according to u'. Consider the collection of copies $\mathcal{D}'_{\text{new}} = \textsc{RetrieveNewCopies}(\mathcal{D}'_{p-1}, H_{\text{last}})$, where the distances in the procedure follow d'. Let A' denote the set of clients whose copies are active in $\mathcal{D}'_{\text{new}}$. Let α'_j be the α-value of the active or indirectly connected copy of client j in $\mathcal{D}'_{\text{new}}$. If no such copy exists, we let $\alpha'_j = 0$, which will not be used later. Let $\pi'(j)$ be the probability that j is directly connected in $\mathcal{D}'_{\text{new}}$. Similarly, we define $d(\cdot, \cdot)$ according to u, and define $(A, \{\alpha_j\}, \{\pi(j)\}, d(\cdot, \cdot))$ according to $\mathcal{D}_{\text{new}} = \textsc{RetrieveNewCopies}(\mathcal{D}_{p-1}, H_{\text{last}})$ under distances d.

As the only difference parameter is $u(\tilde{i}) \in [u'(\tilde{i}) - \eta, u'(\tilde{i})]$, we have $d \leq d'$, meaning it is strictly easier to cross the metric distance thresholds for IC and DC under d. Thus, we can observe that $A \subseteq A'$.

For simplicity, we will consider the viewpoint of Remark 8.3 for η-openability. Based on the fact that h_t was 0-openable with respect to the former setting with $(\tau'_j)_{j \in A'}$, let $(\tau_j)_{j \in A}$ be defined as $\tau_j = \min(\tau'_j, (1-\delta)d^2(j, S'))$ for $j \in A$. Let us check if h_t is η-openable with respect to the latter setting with $(\tau_j)_{j \in A}$ by checking the conditions in Remark 8.3. The first two bullets are straightforward:

- $\tau_j = \alpha_j$ for every $j \in A \setminus B(h_t, \sqrt{\varepsilon\theta})$: Since it was true for $A' \supseteq A$, it is clear that $\tau_j \leq \tau'_j = \alpha_j$. Since $j \in A \subseteq A'$, we have $\alpha_j = \alpha'_j = \tau'_j = \theta$ and $(1-\delta)d^2(j, S') > \alpha_j$, implying $\tau_j \geq \alpha_j$.
- $\alpha_j \leq \tau_j \leq \min\{(1-\delta)d^2(j, S'), (1+\varepsilon^3)\theta\}$ for every $j \in A \cap B(h_t, \sqrt{\varepsilon\theta})$: Fix $j \in A \cap B(h_t, \sqrt{\varepsilon\theta})$. The first inequality holds because $\tau'_j \geq \alpha'_j = \theta = \alpha_j$ and $(1-\delta)d^2(j, S') > \alpha_j$. Since $A \subseteq A'$, $\tau'_j \leq \min\{(1-\delta)d'^2(j, S'), (1+\varepsilon^3)\theta\}$. Since τ_j is defined as $\min(\tau'_j, (1-\delta)d^2(j, S'))$, the second inequality is satisfied.

Before discussing the third and fourth bullets, we establish a useful claim.

Claim 8.9. *Suppose that $\mathcal{H}, \mathcal{DH}$ and $\mathcal{H}, \mathcal{DH}'$ are two partial solutions opening the same set of facilities after phase $p-1$. We assume they are under (f, u) and (f', u') respectively. Suppose*

that u, u' differ on at most one parameter by at most η. Fix any client j. Let α, α' be the α-value of the copies of j that is of type IC or A in $\mathcal{D}_{p-1}$ and $\mathcal{D}'_{p-1}$ (assuming both exist), respectively. Then, $|\alpha - \alpha'| \le \eta$.

Proof. Let $\theta = (1+\varepsilon^3)^{p-1}$. Suppose $\mathcal{H} = (H_1, \ldots, H_{p-1})$ and let $S = \cup_{r\in[p-1]} H_r$. Suppose d, d' are the pairwise distances defined according to u, u', respectively. All the following α-values follow Claim 8.6.

If j has an active copy in both $\mathcal{D}_{p-1}$ and $\mathcal{D}'_{p-1}$, then $\alpha = \alpha' = \theta$.

If j has an indirectly connected copy in both $\mathcal{D}_{p-1}$ and $\mathcal{D}'_{p-1}$, then $\alpha = (1-\delta)d^2(j,S)$ and $\alpha' = (1-\delta)d'^2(j,S)$. Since u, u' differ on at most one parameter by at most η, we have $|d^2(j,S) - d'^2(j,S)| \le \eta$ and thus $|\alpha - \alpha'| \le (1-\delta)\eta \le \eta$.

If j has an active copy in one collection and an indirectly connected copy in the other collection, since $d \le d'$, it must be that j has an active copy in $\mathcal{D}'_{p-1}$ (under d') and an indirectly connected copy in $\mathcal{D}_{p-1}$ (under d). Then, we have $\alpha' = \theta$ and $\alpha = (1-\delta)d^2(j,S) \le \theta$. Since j is active in $\mathcal{D}'_{p-1}$, we have $(1-\delta)d'^2(j,S) > \theta$. Therefore, we have

$$\begin{aligned}|\alpha' - \alpha| &= \theta - (1-\delta)d^2(j,S) \\ &\le (1-\delta)d'^2(j,S) - (1-\delta)d^2(j,S) \le (1-\delta)\eta \le \eta.\end{aligned}$$

□

For the third and fourth bullets, we construct an abstract bid function $bid_{A,\alpha,\tau,\pi,d,\gamma}(j)$ based on the metric openability terms, where $\gamma(j)$ is a client-specific target distance bound:

$$bid_{A,\alpha,\tau,\pi,d,\gamma}(j) = \begin{cases} [\tau_j - \rho\gamma(j)]^+ & \text{if } j \in A, \\ \pi(j)[(1-\delta)d^2(j,S') - \gamma(j)]^+ + (1-\pi(j))[\alpha_j - \rho\gamma(j)]^+ & \text{if } j \notin A \end{cases}$$

For the third bullet, we will set $\gamma(j) = (1-\delta)d^2(h_t, j)$. For the fourth bullet, we will set $\gamma(j) = d^2(i_0, j)$. In both cases, note how the DC term subtracts $\gamma(j)$ directly, while the IC and A terms subtract $\rho\gamma(j)$.

Next, we evaluate $|bid_{A,\alpha,\tau,\pi,d,\gamma}(j) - bid_{A',\alpha',\tau',\pi',d',\gamma}(j)|$ for every client j. Recall that $|\tilde{\pi}(j) - \tilde{\pi}'(j)| \le \eta/M$. We discuss the following cases:

- Suppose that $\tilde{\pi}(j), \tilde{\pi}'(j) > 1 - \eta/M$. Since the procedure RetrieveNewCopies doesn't turn directly connected copies into active or indirectly connected, we have $\pi(j), \pi'(j) > 1 - \eta/M$ and thus $|\pi(j) - \pi'(j)| \le \eta/M$. In addition, we have $j \notin A$ and $j \notin A'$. Because RetrieveNewCopies doesn't change α-values for indirectly connected or active copies, $\alpha_j = (1-\delta)d^2(j,S)$ and $\alpha'_j = (1-\delta)d'^2(j,S)$. Because $d^2(j,S'), d'^2(j,S') \le M$, and $\alpha'_j, \alpha_j \le \theta \le 6M$, we have

$$\begin{aligned}&|bid_{A,\alpha,\tau,\pi,d,\gamma}(j) - bid_{A',\alpha',\tau',\pi',d',\gamma}(j)| \\ &\quad\le 6M \cdot |\pi(j) - \pi'(j)| + \max\{|(1-\delta)d^2(j,S') - (1-\delta)d'^2(j,S')|, |\alpha_j - \alpha'_j|\} \\ &\quad\le 6\eta + \eta = 7\eta.\end{aligned}$$

- Suppose that one of $\tilde{\pi}(j), \tilde{\pi}'(j)$ is strictly less than $1 - \eta/M$. Since $|\tilde{\pi}(j) - \tilde{\pi}'(j)| \le \eta/M$, both are strictly less than 1. This implies j has exactly one active or indirectly connected copy in both $\mathcal{D}_{p-1}$ and $\mathcal{D}'_{p-1}$. We will use bid, bid' for $bid_{A,\alpha,\tau,\pi,d,\gamma}(j)$ and $bid_{A',\alpha',\tau',\pi',d',\gamma}(j)$ respectively for simplicity.
  - If both copies become directly connected, then $\pi(j) = \pi'(j) = 1$ and thus $|bid - bid'| \le |(1-\delta)d^2(j,S') - (1-\delta)d'^2(j,S')| \le \eta$.

– If both copies remain active or indirectly connected, then $\pi(j) = \tilde{\pi}(j), \pi'(j) = \tilde{\pi}'(j)$ and thus $|\pi(j) - \pi'(j)| \le \eta/M$. By Claim 8.9 and the fact that RetrieveNewCopies doesn't modify the α-values of active or indirectly connected copies, we have $|\alpha_j - \alpha'_j| \le \eta$.
  1. If both copies are indirectly connected (i.e., $j \notin A'$), we use the same derivation as above to prove that $|bid - bid'| \le 6\eta + \eta = 7\eta$.
  2. If both copies are active (i.e., $j \in A$), we have $\tau_j \ne \tau'_j$ only if $\tau'_j > (1-\delta)d^2(j, S')$. In this case, we have $\tau_j = (1-\delta)d^2(j, S')$ and $\tau'_j \le (1-\delta)d'^2(j, S')$. Therefore, $|\tau_j - \tau'_j| \le (1-\delta) \cdot |d^2(j, S') - d'^2(j, S')| \le \eta$ and we have $|bid - bid'| \le \eta$.
  3. If $j \in A' \setminus A$ (this is the remaining case because $A \subseteq A'$), we have $\pi'(j) = 0$ and thus $\tilde{\pi}'(j) = 0$. This implies that $\tilde{\pi}(j) \le \eta/M$ and thus $\pi(j) \le \eta/M$. Therefore, $|\pi(j) - \pi'(j)| \le \eta/M$. Furthermore, we have $\tau'_j \ge \theta \ge \alpha_j$, and $\tau'_j \le (1-\delta)d'^2(j, S') \le (1-\delta)(d^2(j, S') + \eta) \le \alpha_j + \eta$ because $j \notin A$. Therefore, we have $|\tau'_j - \alpha_j| \le \eta$. Using the same derivation as above, we have $|bid - bid'| \le 7\eta$.

– If one copy crosses the threshold to become directly connected while the other remains active or indirectly connected, since $d \le d'$, it must be that the copy under d becomes directly connected ($\pi(j) = 1$) while the copy under d' remains active or indirectly connected ($\pi'(j) = \tilde{\pi}'(j)$). We can characterize $bid_{A,\alpha,\tau,\pi,d,\gamma}(j)$ and $bid_{A',\alpha',\tau',\pi',d',\gamma}(j)$ as follows:

$$\begin{aligned} bid_{A,\alpha,\tau,\pi,d,\gamma}(j) &= [(1-\delta)d^2(j, S') - \gamma(j)]^+, \\ bid_{A',\alpha',\tau',\pi',d',\gamma}(j) &= \pi'(j) \cdot [(1-\delta)d'^2(j, S') - \gamma(j)]^+ + (1-\pi'(j))[\alpha'_j - \rho\gamma(j)]^+. \end{aligned} \tag{39}$$

Crucially, because $\pi(j) = 1$ and $|\pi(j) - \pi'(j)| \le \eta/M$, we have $1 - \pi'(j) \le \eta/M$. To bound the difference between the bids, let $X = [(1-\delta)d^2(j, S') - \gamma(j)]^+$, $Y = [(1-\delta)d'^2(j, S') - \gamma(j)]^+$, and $Z = [\alpha'_j - \rho\gamma(j)]^+$. We want to bound $|X - (\pi'(j)Y + (1-\pi'(j))Z)|$. By the triangle inequality, this is at most:

$$\begin{aligned} |X - (\pi'(j)Y + (1-\pi'(j))Z)| &= |X - Y + (1-\pi'(j))(Y - Z)| \\ &\le |X - Y| + (1-\pi'(j))|Y - Z| \ . \end{aligned}$$

First, we bound $|X - Y|$:

$$|X - Y| \le (1-\delta)|d^2(j, S') - d'^2(j, S')| \le \eta \ .$$

Second, we bound $|Y - Z|$. Since all pairwise squared distances are bounded by M, we have $Y \le M$. Furthermore, because the log-adaptive algorithm terminates before θ exceeds $6M$, we have $Z \le \alpha'_j \le \theta \le 6M$. Since both Y and Z are non-negative, their absolute difference is bounded by the maximum of their upper bounds, meaning $|Y - Z| \le 6M$. Therefore,

$$(1-\pi'(j))|Y - Z| \le \frac{\eta}{M} \cdot 6M = 6\eta \ .$$

Putting this back into Equation (39), we conclude that:

$$|bid_{A,\alpha,\tau,\pi,d,\gamma}(j) - bid_{A',\alpha',\tau',\pi',d',\gamma}(j)| \le \eta + 6\eta = 7\eta \ .$$

Using this inequality (which gives a maximum difference of 7η across all cases), we can easily obtain for the sum over all clients:

$$\left|\sum_{j\in D} bid_{A,\alpha,\tau,\pi,d,\gamma}(j) - \sum_{j\in D} bid_{A',\alpha',\tau',\pi',d',\gamma}(j)\right| \leq \sum_{j\in D} 7\eta \leq 7n\eta \ .$$

We check the last two bullets in Remark 8.3 as follows. For the third bullet, we define $\gamma(j) = (1-\delta)d^2(h_t, j)$. Because h_t is a regular facility, $d^2(h_t, j) = d'^2(h_t, j)$. Because h_t was 0-openable with respect to $\theta, f', u', \mathcal{D}'_{p-1}$, the left-hand side under bid' evaluates to at least $\hat{f}$. The difference of at most $7n\eta$ ensures the third bullet is satisfied under $\theta, f, u, \mathcal{D}_{p-1}$ with the required slack bound of $\hat{f}' - 7n\eta \geq \hat{f} - 8n\eta$.

For the fourth bullet, we define $\gamma(j) = d^2(i_0, j)$. With the same abstract bid formulation and distance stability for the regular facility i_0, the sum over all clients under bid' is at most $\hat{f}$. The difference of at most $7n\eta$ ensures the sum under bid is at most $\hat{f} + 7n\eta \leq \hat{f} + 8n\eta$, thereby concluding the fourth bullet. □

Outcome of this section of MergeSolutions. Let

$$\mathcal{H}'', \mathcal{DH}'' = \textsc{CompleteSolution}_{(f,u)}((H'_1, \ldots, H'_p), (\mathcal{D}_0, \ldots, \mathcal{D}_{p-1})) \,,$$

which is a valid call by Claim 8.8. If $\mathcal{H}''$ opens k regular facilities, we return this solution. Otherwise, as $(\mathcal{H}, \mathcal{DH}, f, u)$ and $(\mathcal{H}', \mathcal{DH}', f', u')$ sandwich k, we must have that either $(\mathcal{H}, \mathcal{DH}, f, u)$ and $(\mathcal{H}'', \mathcal{DH}'', f, u)$, or $(\mathcal{H}'', \mathcal{DH}'', f, u)$ and $(\mathcal{H}', \mathcal{DH}', f', u')$ sandwich k.

On the one hand, if $(\mathcal{H}'', \mathcal{DH}'', f, u)$ and $(\mathcal{H}', \mathcal{DH}', f', u')$ sandwich k, then we made progress in that we now have two solutions with a common prefix of length p (instead of $p-1$) and exactly one parameter differs by at most η. However, the new collection of copies $\mathcal{D}''_p$ and $\mathcal{D}'_p$ may not satisfy condition (iv) regarding the bounds on the probabilities of direct connections (π). In this case, we proceed to Subsection 8.3.4 of phase p.

On the other hand, if $(\mathcal{H}, \mathcal{DH}, f, u)$ and $(\mathcal{H}'', \mathcal{DH}'', f, u)$ sandwich k, then these two solutions have the exact same parameters f and u, have $\mathcal{D}_r = \mathcal{D}''_r$ for every $r \leq p-1$, and their first η-valid sequences that differ are H_p and H'_p. In this case, we rename $\mathcal{H}''$ to $\mathcal{H}'$ (and $\mathcal{DH}''$ to $\mathcal{DH}'$) and proceed to the next step.

8.3.2 Maximize Common Prefix of H_p and H'_p

We are given two solutions $(\mathcal{H}, \mathcal{DH}, f, u)$ and $(\mathcal{H}', \mathcal{DH}', f, u)$ on the same parameters f, u that sandwich k. It is guaranteed that $\forall r \leq p-1, \mathcal{D}_r = \mathcal{D}'_r$ in the history of collections of copies. Moreover, if we let $\mathcal{H} = (H_1, \ldots, H_L)$ and $\mathcal{H}' = (H'_1, \ldots, H'_{L'})$, then $H_i = H'_i$ for $i = 1, \ldots, p-1$ and $H_p \neq H'_p$. In addition, H_p and H'_p only contain regular facilities.

In this step of MergeSolutions, we increase the common prefix of H_p and H'_p by "slowly" modifying H'_p. As H_p only contains regular facilities, we let $H_p = \langle i_1, i_2, \ldots, i_\ell \rangle$. Moreover, as the parameters (f, u) and $\mathcal{D}_1, \ldots, \mathcal{D}_{p-1}$ are clear from the context, we omit them in the following when, e.g., using the definition of η-openability, and calling the procedures CompleteSolution and CompleteSequence.[7]

This subsection proceeds similarly to the approach in [CCG+25].

[7] When we use CompleteSolution $(H_1, \ldots, H_{p-1}, H''_p)$, we are actually referring to CompleteSolution $(\mathcal{H}'' = (H_1, \ldots, H_{p-1}, H''_p), \mathcal{DH}'' = (\mathcal{D}_0, \ldots, \mathcal{D}_{p-1}))$. When we use CompleteSequence (H''_p), we are actually referring to CompleteSequence $(\mathcal{H}'' = (H_1, \ldots, H_{p-1}), H''_p, \mathcal{DH}'' = (\mathcal{D}_0, \ldots, \mathcal{D}_{p-1}))$.

Transforming H'_p step-by-step so that H_p becomes a prefix.

Repeat the following until all of H_p becomes a prefix of H'_p.

- Let q be the largest index so that $H'_p = \langle i_1, i_2, \ldots, i_q, h'_{q+1}, \ldots, h'_{\ell_1} \rangle$, i.e., q equals the length of the common prefix $i_1, i_2, \ldots, i_q$ of H_p and H'_p.
- Update H'_p by inserting a free copy of i_{q+1}, denoted $\tilde{i}_{q+1}$, at the end of H'_p with $u(\tilde{i}_{q+1}) = 0$. That is, we get the sequence

$$H'_p = \langle i_1, i_2, \ldots, i_q, h'_{q+1}, \ldots, h'_{\ell_1}, \tilde{i}_{q+1} \rangle.$$

- For $t = q+1, \ldots, \ell_1$,
  - Update H'_p by first removing h'_t and then completing the sequence by a call to CompleteSequence. More explicitly, if

$$H'_p = \langle i_1, \ldots, i_q, h'_t, h'_{t+1} \ldots, h'_{\ell_1}, \tilde{i}_{q+1}, h'_{\ell_1+1}, \ldots, h'_{\ell_2} \rangle$$

    then we delete h'_t to obtain

$$H''_p = \langle i_1, \ldots, i_q, h'_{t+1} \ldots, h'_{\ell_1}, \tilde{i}_{q+1}, h'_{\ell_1+1}, \ldots, h'_{\ell_2} \rangle$$

    and finally update $H'_p = \text{CompleteSequence}(H''_p)$ which will be of the form

$$\langle i_1, \ldots, i_q, h'_{t+1}, \ldots, h'_{\ell_1}, \tilde{i}_{q+1}, h'_{\ell_1+1}, \ldots, h'_{\ell_2}, \ldots, h'_{\ell_3} \rangle,$$

    where $h'_{\ell_2+1}, \ldots, h'_{\ell_3}$ are the facilities added by CompleteSequence (which could be empty).
- After removing $h'_{q+1}, \ldots, h'_{\ell_1}$, we have

$$H'_p = \langle i_1, i_2, \ldots, i_q, \tilde{i}_{q+1}, h'_x, \ldots, h'_y \rangle.$$

  We update H'_p by replacing $\tilde{i}_{q+1}$ with its regular copy i_{q+1}, i.e., we obtain

$$H'_p = \langle i_1, i_2, \ldots, i_q, i_{q+1}, h'_x, \ldots, h'_y \rangle.$$

We have the following claim.

Claim 8.10. *Throughout the updates to H'_p, $H'_1, \ldots, H'_p$ remain η-valid sequences, and H'_p contains at most one free facility.*

Proof. As we never update f or the value $u(\cdot)$ of a free facility in $H'_1, \ldots, H'_{p-1}$, these sequences clearly remain η-valid.

We continue to argue that H'_p remains η-valid throughout its updates. By Claim 8.8, this is true before any updates to H'_p. We now analyze the three types of updates to H'_p.

- Updating H'_p by inserting a free copy of i_{q+1} maintains that it is η-valid as every regular facility stays η-openable in the sequence (they have the same prefix as before) and the sequence remains maximal.

- Updating H'_p by removing h'_t and calling CompleteSequence. Removing a h'_t leaves the new prefix as a subsequence of the old prefix. Thus, each subsequent regular facility is still η-openable with respect to the original sequence padded with h'_t (which acts as a valid super-sequence $\mathcal{H}'$). Finally, the call to CompleteSequence adds facilities until no more can be opened, ensuring that the sequence is maximal.
- Updating H'_p by replacing $\tilde{i}_{q+1}$ with its regular copy i_{q+1}. First note that, since $u(\tilde{i}_{q+1}) = 0$, this does not impact whether the following regular facilities are η-openable (with respect to a super-sequence), as the distances $d^2(\tilde{i}_{q+1}, \cdot)$ and $d^2(i_{q+1}, \cdot)$ are perfectly identical. By the same argument, the sequence stays maximal. Finally, to see that i_{q+1} is η-openable (with respect to a potential super-sequence), notice that the prefix $i_1, \ldots, i_q$ is the same as that in H_p, which is already known to be an η-valid sequence.

We complete the proof of the claim by arguing that H'_p contains at most one free facility at any point in time. This follows from the facts that (1) H'_p initially contains no free facilities, (2) in one repetition, the single free facility $\tilde{i}_{q+1}$ is inserted into H'_p, and (3) $\tilde{i}_{q+1}$ is strictly replaced by its regular copy i_{q+1} before continuing to the next repetition. □

Outcome of this section of MergeSolutions. We distinguish three cases.

Case 1. If at some point H'_p is such that CompleteSolution$(H'_1, H'_2, \ldots, H'_p)$ opens exactly k facilities, then we return that solution.

Case 2. Otherwise, suppose there is an update to H'_p so that if we let H_p^{before} and H_p^{after} denote H'_p before and after this update, respectively, then CompleteSolution$(H_1, H_2, \ldots, H_p^{\text{before}})$ and CompleteSolution$(H_1, H_2, \ldots, H_p^{\text{after}})$ sandwich k. There are three kinds of updates to H'_p. However, note that the update where $\tilde{i}_{q+1}$ is replaced by its regular copy i_{q+1} cannot satisfy the conditions of this case. Indeed, as $u(\tilde{i}_{q+1}) = 0$, we have

$$\text{CompleteSolution}(H_1, \ldots, H_p^{\text{before}}) = \text{CompleteSolution}(H_1, \ldots, H_p^{\text{after}})$$

in this case, which contradicts that they sandwich k.

We proceed to analyze the two other kinds of updates. If the update to H'_p is to insert a free facility $\tilde{i}_{q+1}$ with $u(\tilde{i}_{q+1}) = 0$, then

$$\begin{aligned} H_p^{\text{before}} &= \langle i_1, i_2, \ldots, i_q, h'_{q+1}, \ldots, h'_{\ell_1} \rangle \\ H_p^{\text{after}} &= \langle i_1, i_2, \ldots, i_q, h'_{q+1}, \ldots, h'_{\ell_1}, \tilde{i}_{q+1} \rangle \end{aligned}$$

Note that the solution CompleteSolution$(H_1, H_2, \ldots, H_p^{\text{before}})$ is equivalent to CompleteSolution$(H_1, H_2, .$ if we set $u(\tilde{i}_{q+1}) = 10M$ (except for the free facility $\tilde{i}_{q+1}$). We can thus do binary search on $u(\tilde{i}_{q+1})$ to find two values $u'(\tilde{i}_{q+1})$ and $u(\tilde{i}_{q+1})$ with $|u'(\tilde{i}_{q+1}) - u(\tilde{i}_{q+1})| \leq \eta$ so that the solutions CompleteSolution$_{(f,u')}(H_1, \ldots, H_p^{\text{after}})$ and CompleteSolution$_{(f,u)}(H_1, \ldots, H_p^{\text{after}})$ sandwich k (where u and u' are identical except for $\tilde{i}_{q+1}$). We can thus proceed to Subsection 8.3.4 of phase p, since they have the first p, instead of $p-1$, η-valid sequences in common and exactly one difference parameter. Their maximality follows from that of H_p^{before}.

Finally, consider when the update to H'_p is of the second type, i.e., $u(\tilde{i}_{q+1}) = 0$ and

$$\begin{aligned} H_p^{\text{before}} &= \langle i_1, \ldots, i_q, h'_t, h'_{t+1} \ldots, h'_{\ell_1}, \tilde{i}_{q+1}, h'_{\ell_1+1}, \ldots, h'_{\ell_2} \rangle \\ H_p^{\text{after}} &= \langle i_1, \ldots, i_q, h'_{t+1}, \ldots, h'_{\ell_1}, \tilde{i}_{q+1}, h'_{\ell_1+1}, \ldots, h'_{\ell_2}, \ldots, h'_{\ell_3} \rangle \end{aligned}$$

In this case, CompleteSolution$(H_1, \ldots, H_p^{\text{before}})$ and CompleteSolution$(H_1, \ldots, H_p^{\text{after}})$ are two solutions with the same parameters f, u, that sandwich k and we proceed to the next step with the following properties of H_p^{before} and H_p^{after}:

- Both sequences contain one free facility, $\tilde{i}_{q+1}$.
- H_p^{before} and H_p^{after} are identical except that (1) H_p^{before} contains a regular facility h'_t not present in H_p^{after}, and (2) H_p^{after} contains a (potentially empty) suffix of regular facilities not present in H_p^{before}.

Case 3. None of the previous cases apply. Then, completing the solution with the initial H'_p and the final H'_p cannot sandwich k. Since then (assuming the first case does not apply) there must be an update to H'_p that results in two solutions that sandwich k, since the completion of H_p and the completion of the initial H'_p sandwich k. The above implies that if we let $H_p^{\text{before}} = H_p$ and H_p^{after} equal the final H'_p, then CompleteSolution$(H_1, \ldots, H_p^{\text{before}})$ and CompleteSolution$(H_1, \ldots, H_p^{\text{after}})$ sandwich k. Further note that since H_p^{after} equals H'_p after the final updates, H_p^{before} appears as a prefix of H_p^{after} and both sequences only contain regular facilities. To summarize, in this case, CompleteSolution$(H_1, \ldots, H_p^{\text{before}})$ and CompleteSolution$(H_1, \ldots, H_p^{\text{after}})$ are two solutions with the same parameters f, u, that sandwich k and we proceed to the next step with the following properties of H_p^{before} and H_p^{after}:

- Both sequences contain no free facilities.
- H_p^{before} and H_p^{after} are identical except that H_p^{after} contains a suffix of regular facilities not present in H_p^{before}.

8.3.3 Removing Extra Facilities

This subsection proceeds almost identically to that of [CCG+25] (except that we will go to the next subroutine instead of finishing phase p). As in Subsection 8.3.2, the parameters (f, u) and $\mathcal{D}_1, \ldots, \mathcal{D}_{p-1}$ are clear from the context, and we will omit them when using the definition of η-openability and calling the function CompleteSolution.

The input to this step consists of two solutions CompleteSolution$(H_1, \ldots, H_p^{\text{before}})$ and CompleteSolution$(H_1, \ldots, H_p^{\text{after}})$ that have the same parameters f, u and sandwich k. Moreover, H_p^{before} and H_p^{after} both contain at most one free facility and they are identical except for two potential differences:

- H_p^{before} may contain a facility not present in H_p^{after};
- H_p^{after} contains a suffix of regular facilities (which may be empty) that is not present in H_p^{before}.

To simplify notation in this section, we let i_* denote the regular facility in H_p^{before} that is not present in H_p^{after} if it exists (so i_* corresponds to h'_t in the previous section). We further let $h_1, \ldots, h_\ell$ denote the suffix of H_p^{after} not present in H_p^{before}.

We now transform H_p^{after} into H_p^{before} one change at a time:

Generation of η-valid sequences from H_p^{after} to H_p^{before}.

- If H_p^{before} contains the regular facility i_* not present in H_p^{after}, then let $\tilde{i}_*$ be a free copy of i_* with $u(\tilde{i}_*) = 0$ and obtain H_p^0 from H_p^{after} by adding $\tilde{i}_*$ at the end. Otherwise, we let H_p^0 equal H_p^{after}.
- For $r = 1, \ldots, \ell$, obtain H_p^r from H_p^{r-1} by removing h_r.

We remark that H_p^ℓ is identical to H_p^{before} since we removed the whole suffix $h_1, \ldots, h_\ell$ present in H_p^{after}, except for potentially the facility i_*. That is, the only difference in H_p^{before} and H_p^ℓ may be the placement of $\tilde{i}_*$ and that $\tilde{i}_*$ is a free copy of the regular facility i_* present in H_p^{before}. However, since $u(\tilde{i}_*) = 0$, we have that the two sequences H_p^{before} and H_p^ℓ have an identical impact on CompleteSolution, which is a fact we use in the proof of Claim 8.12.

Claim 8.11. *For $r = 0, 1 \ldots \ell$, H_p^r is an η-valid sequence with at most two free facilities.*

Proof. The sequence H_p^r has at most two free facilities because H_p^{after} has at most one free facility, and we potentially add one new free facility $\tilde{i}_*$.

We now verify that H_p^r is η-valid. For any regular facility h' in the sequence, the prefix of H_p^r before h' is a subsequence of the prefix before h' in H_p^{after}. Thus, as h' was openable with respect to a super-sequence $\mathcal{H}(h')$ in H_p^{after}, h' must be η-openable with respect to the same super-sequence $\mathcal{H}(h')$ in H_p^r. This ensures that the first property of Definition 8.4 is satisfied. To verify maximality, notice that since we added $\tilde{i}_*$ with $u(\tilde{i}_*) = 0$ at the end, we have that the set of active clients after opening up the facilities in H_p^r is a subset of the active clients after opening up the facilities in H_p^{before}. Hence, the maximality of H_p^{before} implies the maximality of H_p^r. □

Claim 8.12. *One of the following is true.*

- *For one of the sequences H_p^r, CompleteSolution$(H_1, H_2, \ldots, H_p^r)$ opens k regular facilities.*
- *CompleteSolution$(H_1, H_2, \ldots, H_p^{after})$ and CompleteSolution$(H_1, H_2, \ldots, H_p^0)$ sandwich k.*
- *CompleteSolution$(H_1, H_2, \ldots, H_p^r)$ and CompleteSolution$(H_1, H_2, \ldots, H_p^{r+1})$ sandwich k for an $r \in \{0, 1, \ldots, \ell - 1\}$.*

Proof. Assume the first bullet point does not hold, i.e., the completion of no H_p^r leads to the opening of k regular facilities. Now notice that the second and third bullet points ask whether consecutive solutions sandwich k, starting with H_p^{after} and ending with H_p^ℓ. By definition, H_p^ℓ equals H_p^{before} except potentially for the placement of $\tilde{i}_*$ and that $\tilde{i}_*$ is the free copy of i_* present in H_p^{before}. However, as $u(\tilde{i}_*) = 0$, the two sequences have an identical impact when we complete the solution, which implies

$$\text{CompleteSolution}(H_1, \ldots, H_p^\ell) = \text{CompleteSolution}(H_1, \ldots, H_p^{\text{before}}).$$

Thus, as CompleteSolution$(H_1, \ldots, H_p^{\text{after}})$ and CompleteSolution$(H_1, \ldots, H_p^{\text{before}})$ sandwich k, either the second or third bullet point of the statement must be valid (since no solution opened exactly k regular facilities). □

Outcome of this section of MergeSolutions. We distinguish three cases based on the first bullet point that is true in the above claim.

Case 1. If one of the sequences H_p^r is such that $\textsc{CompleteSolution}(H_1, H_2, \ldots, H_p^r)$ opens k regular facilities, then we return that solution.

Case 2. Otherwise, if the two solutions $\textsc{CompleteSolution}(H_1, H_2, \ldots, H_p^{\text{after}})$ and $\textsc{CompleteSolution}(H_1, H_2, \ldots, H_p^0)$ sandwich k, then H_p^0 is equal to H_p^{after} except that $\tilde{i}_*$ was added at the end of H_p^{after} with $u(\tilde{i}_*) = 0$. Note that the solution $\textsc{CompleteSolution}(H_1, H_2, \ldots, H_p^{\text{after}})$ is equivalent to $\textsc{CompleteSolution}(H_1, H_2, \ldots, H_p^0)$ if we set $u(\tilde{i}_*) = 10M$ (except for the presence of $\tilde{i}_*$). We can thus do a binary search on $u(\tilde{i}_*)$ to find two values $u'(\tilde{i}_*)$ and $u(\tilde{i}_*)$ with $|u'(\tilde{i}_*) - u(\tilde{i}_*)| \leq \eta$ so that the two solutions $\textsc{CompleteSolution}_{(f,u')}(H_1, \ldots, H_p^0)$ and $\textsc{CompleteSolution}_{(f,u)}(H_1, \ldots, H_p^0)$ sandwich k (where u and u' are identical except for $\tilde{i}_*$). We can thus proceed to Subsection 8.3.4 of phase p, since they have the first p, instead of $p-1$, η-valid sequences in common and exactly one difference parameter.

Case 3. Finally, consider the case when none of the above cases apply and (thus by the above claim) there is an $r \in \{0, 1, \ldots, \ell - 1\}$ so that $\textsc{CompleteSolution}(H_1, H_2, \ldots, H_p^r)$ and $\textsc{CompleteSolution}(H_1, H_2, \ldots, H_p^{r+1})$ sandwich k. The only difference between H_p^r and H_p^{r+1} is that H_p^r contains h_{r+1} which is not present in H_p^{r+1}. Let H_p be the sequence obtained by adding a free copy $\tilde{i}_{r+1}$ of h_{r+1} at the end of H_p^{r+1} (which clearly maintains that it is η-valid), then

$$\textsc{CompleteSolution}(H_1, H_2, \ldots, H_p) = \textsc{CompleteSolution}(H_1, H_2, \ldots, H_p^r) \text{ if } u(\tilde{i}_{r+1}) = 0$$

and

$$\textsc{CompleteSolution}(H_1, H_2, \ldots, H_p) = \textsc{CompleteSolution}(H_1, H_2, \ldots, H_p^{r+1}) \text{ if } u(\tilde{i}_{r+1}) = 10M.$$

Hence, doing a binary search on $u(\tilde{i}_{r+1})$ as in the previous case can proceed to Subsection 8.3.4 of phase p, since they have the first p, instead of $p-1$, η-valid sequences in common and exactly one difference parameter.

8.3.4 Making the Probability of Direct Connects Almost Identical

The input of this step consists of solutions $(\mathcal{H}, \mathcal{DH}, f, u)$ and $(\mathcal{H}, \mathcal{DH}', f', u')$ such that (f, u) and (f', u') have at most one difference parameter. Moreover, the two solutions sandwich k. However, $\mathcal{D}_p$ and $\mathcal{D}'_p$ are simply obtained from $\mathcal{D}_{p-1}$ and $\mathcal{D}'_{p-1}$ respectively by running CompleteCopies. In this step, we will walk between the collection of copies $\mathcal{D}_p$ and $\mathcal{D}'_p$ following Definition 8.5 so that condition (iv) holds for phase p.

We follow the procedure below twice. Suppose that $S = \cup_{r \in [p]} H_r$. Suppose that d, d' are the pairwise distances induced by u, u'. One run is to transform $(\mathcal{D}_p, \mathcal{D}_{p-1}, d)$ towards $(\mathcal{D}'_p, \mathcal{D}'_{p-1}, d')$. The other run is to transform $(\mathcal{D}'_p, \mathcal{D}'_{p-1}, d')$ towards $(\mathcal{D}_p, \mathcal{D}_{p-1}, d)$.

Transforming connection types in $(\mathcal{D}, \tilde{\mathcal{D}}, d)$ towards $(\mathcal{D}', \tilde{\mathcal{D}}', d')$

Repeat the following.

- Find a copy j in $\mathcal{D}$ and j' in $\mathcal{D}'$ such that
  - j and j' correspond to the same client,
  - j and j' are both of type IC or A in $\tilde{\mathcal{D}}$ and $\tilde{\mathcal{D}}'$,
  - j is of type DC in $\mathcal{D}$ while j' is of type IC or A in $\mathcal{D}'$.

  If no such client exists, then stop the process.
- Update $\mathcal{D}$ by changing j to IC type and update the α-value α_j to $(1-\delta)d^2(j,S)$.

After running the two procedures, we can obtain a sequence of solutions $\tilde{\mathcal{D}}_0 = \mathcal{D}_p, \tilde{\mathcal{D}}_1, \ldots, \tilde{\mathcal{D}}_m$ by the first run, while another sequence of solutions $\tilde{\mathcal{D}}'_0 = \mathcal{D}'_p, \tilde{\mathcal{D}}'_1, \ldots, \tilde{\mathcal{D}}'_{m'}$ is obtained by the second run. First, we show that all $\tilde{\mathcal{D}}_r$ and $\tilde{\mathcal{D}}'_r$ are valid collections of copies.

Claim 8.13. *Consider time $\theta = (1+\varepsilon^3)^{p-1}$. For any $r \in [m]$, $\tilde{\mathcal{D}}_r$ is valid with respect to $\theta, (H_1, \ldots, H_p), \mathcal{D}_{p-1}$. For any $r \in [m']$, $\tilde{\mathcal{D}}'_r$ is valid with respect to $\theta, (H'_1, \ldots, H'_p), \mathcal{D}'_{p-1}$.*

Proof. We prove the claim for $\tilde{\mathcal{D}}_r$; the proof for $\tilde{\mathcal{D}}'_r$ is symmetric. We proceed by induction on r.

For the base case when $r = 0$, because $\tilde{\mathcal{D}}_0 = \mathcal{D}_p$, by the definition of CompleteCopies, we have that $\tilde{\mathcal{D}}_0$ is perfectly valid with respect to $\theta, (H_1, \ldots, H_p), \mathcal{D}_{p-1}$.

Next, we consider the induction step when $r > 0$. Suppose $\tilde{\mathcal{D}}_{r-1}$ is valid with respect to $\theta, (H_1, \ldots, H_p), \mathcal{D}_{p-1}$. In the procedure, we update $\tilde{\mathcal{D}}_{r-1}$ to $\tilde{\mathcal{D}}_r$ by changing a copy j from type DC to type IC. To show that $\tilde{\mathcal{D}}_r$ is valid with respect to $\theta, (H_1, \ldots, H_p), \mathcal{D}_{p-1}$, we need to show that its original α-value in $\mathcal{D}_{p-1}$ satisfies the slack condition: $\rho(1-\delta)d^2(j,S) \le \alpha_j \le \rho(1-\delta)d^2(j,S) + 4\eta$.

First, because j was IC or A in $\mathcal{D}_{p-1}$ but became DC in $\mathcal{D}_p$, it must have triggered the DC conversion during RetrieveNewCopies against the newly opened facilities H_p. Therefore, it strictly satisfied $\alpha_j \ge \rho(1-\delta)d^2(j,H_p)$. Since $S = S_{\text{old}} \cup H_p$, we trivially have $d^2(j,S) \le d^2(j,H_p)$, satisfying the lower bound $\alpha_j \ge \rho(1-\delta)d^2(j,S)$.

Second, suppose that j' is the corresponding copy of j in $\mathcal{D}'_{p-1}$ (used in the above procedure), with dual value $\alpha'_{j'}$. Since j, j' are of type IC or A in $\mathcal{D}_{p-1}, \mathcal{D}'_{p-1}$ respectively, and u, u' differ in at most one parameter by at most η, Claim 8.9 gives $|\alpha_j - \alpha'_{j'}| \le \eta$.

To bound $\alpha'_{j'}$, we look at its behavior under d'. Because j' was IC or A in the valid collection $\mathcal{D}'_{p-1}$, its value is exactly $(1-\delta)d'^2(j,S_{\text{old}})$ or θ (with $\theta < (1-\delta)d'^2(j,S_{\text{old}})$). In either case, since $\rho > 1$, it strictly satisfies $\alpha'_{j'} \le \rho(1-\delta)d'^2(j,S_{\text{old}})$. Furthermore, since it remains of type IC or A in $\mathcal{D}'_p$, it did *not* trigger the RetrieveNewCopies conversion under d' against H_p, meaning $\alpha'_{j'} < \rho(1-\delta)d'^2(j,H_p)$. Combining these bounds against S_{old} and H_p, we establish its bound against the full set $S = S_{\text{old}} \cup H_p$:

$$\alpha'_{j'} \le \rho(1-\delta)\min(d'^2(j,S_{\text{old}}), d'^2(j,H_p)) = \rho(1-\delta)d'^2(j,S)\ .$$

Finally, because u and u' differ by at most η, we have $d'^2(j,S) \le d^2(j,S) + \eta$. Therefore, we can tightly bound α_j against d:

$$\begin{aligned}
\alpha_j \le \alpha'_{j'} + \eta &\le \rho(1-\delta)d'^2(j,S) + \eta \\
&\le \rho(1-\delta)(d^2(j,S) + \eta) + \eta \\
&= \rho(1-\delta)d^2(j,S) + \rho(1-\delta)\eta + \eta \\
&\le \rho(1-\delta)d^2(j,S) + (1+\rho)\eta \\
&\le \rho(1-\delta)d^2(j,S) + 4\eta\ .
\end{aligned}$$

Therefore, the original α_j value securely satisfies the required 4η validity bounds. After updating j's type to IC and setting its new α-value to $(1-\delta)d^2(j,S)$, all conditions in Definition 8.5 hold for $\tilde{\mathcal{D}}_r$. This completes the induction. □

Second, we show that $\tilde{\mathcal{D}}_m$ and $\tilde{\mathcal{D}}'_{m'}$ are close enough so that condition (iv) holds for them.

Claim 8.14. *Fix any client j. Suppose $\pi(j), \pi'(j)$ are the probabilities of direct connects for j in $\tilde{\mathcal{D}}_m$ and $\tilde{\mathcal{D}}'_{m'}$ respectively. Then, $|\pi(j) - \pi'(j)| \le \eta/M$.*

Proof. Because client j has at most one copy of type IC or A in both $\tilde{\mathcal{D}}_m$ and $\tilde{\mathcal{D}}'_{m'}$, we can use the mass of this unique IC or A copy (if any) to determine the exact DC probabilities $\pi(j)$ and $\pi'(j)$. Suppose $\tilde{\pi}(j), \tilde{\pi}'(j)$ are the probabilities of direct connects for j in $\mathcal{D}_{p-1}$ and $\mathcal{D}'_{p-1}$ respectively. Let $\tilde{j}$ and $\tilde{j}'$ be the unique copies of type IC or A for client j in $\mathcal{D}_{p-1}$ and $\mathcal{D}'_{p-1}$ respectively (if they exist).

If one of them does not exist (meaning the client is fully DC in that historical state), because we have the invariant $|\tilde{\pi}(j) - \tilde{\pi}'(j)| \le \eta/M$, we must have $\tilde{\pi}(j), \tilde{\pi}'(j) \ge 1 - \eta/M$. The walking procedure only acts to revert copies that were IC/A in the previous phase; it does not convert historical DC copies. Thus, the historical DC mass is permanent, meaning $\pi(j) \ge \tilde{\pi}(j)$ and $\pi'(j) \ge \tilde{\pi}'(j)$. This mathematically restricts both final probabilities to the top of the interval $[1 - \eta/M, 1]$, strictly ensuring $|\pi(j) - \pi'(j)| \le \eta/M$.

Next, we consider the case when both $\tilde{j}$ and $\tilde{j}'$ exist. Because we run the bidirectional walking procedures until no more possible updates can be made, it is impossible for $\tilde{j}$ and $\tilde{j}'$ to end up with divergent types (one DC and one IC/A). Therefore, we have only two possible atomic outcomes for these specific copies:

1. They both evaluate to type DC in $\tilde{\mathcal{D}}_m$ and $\tilde{\mathcal{D}}'_{m'}$. Because $\tilde{j}$ and $\tilde{j}'$ hold the entirety of the non-DC mass (exactly $1 - \tilde{\pi}(j)$ and $1 - \tilde{\pi}'(j)$ respectively), converting them to DC forces the total DC probability to $\tilde{\pi}(j) + (1 - \tilde{\pi}(j)) = 1$. Thus, $\pi(j) = \pi'(j) = 1$, yielding a difference of 0.

2. They both evaluate to type IC/A, meaning the procedure reverted any DC conversions from this phase. In this case, the DC probabilities remain exactly their historical values: $\pi(j) = \tilde{\pi}(j)$ and $\pi'(j) = \tilde{\pi}'(j)$. By our inductive promise, their difference is bounded by η/M.

In all cases, we strictly maintain the invariant $|\pi(j) - \pi'(j)| \le \eta/M$. □

The outcome of this section of MergeSolutions. Because the previous subroutines successfully aligned the opened facilities up to phase p, we have $H'_i = H_i$ for all $i \le p$. We can

thus obtain a sequence of solutions as follows:

$$\begin{aligned}
\mathcal{H}^{(0)}, \mathcal{DH}^{(0)} &= \textsc{CompleteSolution}_{f,u}((H_1, \ldots, H_p), (\mathcal{D}_0, \ldots, \mathcal{D}_{p-1}, \tilde{\mathcal{D}}_0)) \\
\mathcal{H}^{(1)}, \mathcal{DH}^{(1)} &= \textsc{CompleteSolution}_{f,u}((H_1, \ldots, H_p), (\mathcal{D}_0, \ldots, \mathcal{D}_{p-1}, \tilde{\mathcal{D}}_1)) \\
&\vdots \\
\mathcal{H}^{(m)}, \mathcal{DH}^{(m)} &= \textsc{CompleteSolution}_{f,u}((H_1, \ldots, H_p), (\mathcal{D}_0, \ldots, \mathcal{D}_{p-1}, \tilde{\mathcal{D}}_m)) \\
\mathcal{H}^{(m+1)}, \mathcal{DH}^{(m+1)} &= \textsc{CompleteSolution}_{f',u'}((H_1, \ldots, H_p), (\mathcal{D}'_0, \ldots, \mathcal{D}'_{p-1}, \tilde{\mathcal{D}}'_{m'})) \\
\mathcal{H}^{(m+2)}, \mathcal{DH}^{(m+2)} &= \textsc{CompleteSolution}_{f',u'}((H_1, \ldots, H_p), (\mathcal{D}'_0, \ldots, \mathcal{D}'_{p-1}, \tilde{\mathcal{D}}'_{m'-1})) \\
&\vdots \\
\mathcal{H}^{(m+m'+1)}, \mathcal{DH}^{(m+m'+1)} &= \textsc{CompleteSolution}_{f',u'}((H_1, \ldots, H_p), (\mathcal{D}'_0, \ldots, \mathcal{D}'_{p-1}, \tilde{\mathcal{D}}'_0))
\end{aligned}$$

It is clear that the first one and the last one sandwich k. Therefore, there must be two consecutive solutions in the above sequence that sandwich k. We can discuss two cases then.

Case 1. If $(\mathcal{H}^{(m)}, \mathcal{DH}^{(m)})$ and $(\mathcal{H}^{(m+1)}, \mathcal{DH}^{(m+1)})$ sandwich k, according to Claim 8.13 and Claim 8.14, we can proceed to phase $p+1$ of MergeSolutions with these two solutions. Since the connection types in $\tilde{\mathcal{D}}_m$ and $\tilde{\mathcal{D}}'_{m'}$ are perfectly aligned, the only difference between the two solutions is the original difference parameter (either $f \neq f'$ or $u \neq u'$). They share the first p sequences in common, satisfy condition (iv), and differ by exactly one difference parameter.

Case 2. Suppose that $(\mathcal{H}^{(r)}, \mathcal{DH}^{(r)})$ and $(\mathcal{H}^{(r+1)}, \mathcal{DH}^{(r+1)})$ sandwich k for some $r \neq m$. Without loss of generality, we assume that $r < m$ (the other case is symmetric). Suppose j is the copy whose connection type differs in $\tilde{\mathcal{D}}_r$ and $\tilde{\mathcal{D}}_{r+1}$, where j is of type DC in $\tilde{\mathcal{D}}_r$ while j is of type IC in $\tilde{\mathcal{D}}_{r+1}$. We observe that condition (iv) only fails on the client corresponding to copy j.

We can further split this copy j by choosing $\mu' \in [0, \mu(j)]$ such that one copy $j^{(1)}$ has IC type with probability μ' and the other copy $j^{(2)}$ has DC type with probability $\mu(j) - \mu'$. We denote the resulting collection of copies by $\tilde{\mathcal{D}}^{\text{sp}}(\mu')$. According to Claim 8.13, because $\tilde{\mathcal{D}}_r$ and $\tilde{\mathcal{D}}_{r+1}$ are valid collections of copies, any interpolation $\tilde{\mathcal{D}}^{\text{sp}}(\mu')$ is a valid collection of copies. Therefore, we can obtain a solution $(\mathcal{H}^{\text{sp}}(\mu'), \mathcal{DH}^{\text{sp}}(\mu'))$ by calling $\textsc{CompleteSolution}_{f,u}((H_1, \ldots, H_p), (\mathcal{D}_0, \ldots, \mathcal{D}_{p-1}, \tilde{\mathcal{D}}^{\text{sp}}(\mu')))$.

By binary searching μ', we can find two values μ'_1 and μ'_2 with $|\mu'_1 - \mu'_2| \leq \eta/M$ in polynomial time such that the two solutions $(\mathcal{H}^{\text{sp}}(\mu'_1), \mathcal{DH}^{\text{sp}}(\mu'_1))$ and $(\mathcal{H}^{\text{sp}}(\mu'_2), \mathcal{DH}^{\text{sp}}(\mu'_2))$ sandwich k.

Because the overall probabilities of direct connections for client j in the two resultant solutions $(\mathcal{DH}^{\text{sp}}(\mu'_1))_p$ and $(\mathcal{DH}^{\text{sp}}(\mu'_2))_p$ differ by exactly $|\mu'_1 - \mu'_2|$, and we bounded this difference by η/M, these two solutions satisfy condition (iv). We can then proceed to phase $p+1$ of MergeSolutions with these two solutions: they have the first p η-valid sequences in common, utilize identical parameters (f, u), and the newly established parameter between them is exactly the connection probability mass μ' (which is not viewed as a difference parameter).

8.4 Analysis of the Robust Algorithm

In this subsection, we prove Theorem 8.7, restated as follows, showing that a solution $(\mathcal{H} = (H_1, \ldots, H_L), \mathcal{DH} = (\mathcal{D}_0, \mathcal{D}_1, \ldots, \mathcal{D}_L))$ consisting of η-valid sequences and valid collections of

copies yields an almost LMP Γ-approximation. This subsection follows the same logic as Subsection 5.4 from [CCG+26], the main difference is in establishing the constraints for the Dual Payment, which are different for this new algorithm.

Theorem 8.7. *Consider a solution $\mathcal{H}, \mathcal{DH}$ of η-valid sequences and valid collections of copies. Let S be the opened facilities. Further, let S_{free} and $S_{\mathsf{reg}} = S - S_{\mathsf{free}}$ be the free copies and regular facilities, respectively. Then, we have*

$$\sum_{j \in D} d^2(j, S) \leq \frac{\Gamma + 201\sqrt{\varepsilon} + 20n\eta}{1 - \delta} \cdot (opt_{LP}(f) - |S_{\mathsf{reg}}| \cdot f) \ .$$

where

- *if d is a (general) metric, we choose $\rho = 2.5, \Gamma = 4.9$; or*
- *if d is a Euclidean metric, we choose $\rho = 2, \Gamma = 3 + \ln(2)$.*

Our analysis closely follows that of LogAdaptiveAlgorithm in Section 7. However, one key difference is that in the current setting, each client $j \in D$ may have multiple copies in its collection of copies D_j. Older setting in Section 7 can be seen as a special case where each client has only one weighted copy. To close the gap between the two settings, for each copy $\tilde{j} \in \mathcal{D}_L$ in the final collection of copies, it induces a corresponding weighted client with weight $\mu(\tilde{j})$, which is equal to the probability mass of the copy $\tilde{j}$ in $\mathcal{D}_L$. For simplicity, in the rest of this subsection, we will call these induced clients by the copies simply as "clients" and call the clients in D as "original clients". We will simply use j to denote such a client, and use $oc(j)$ to denote its corresponding "original client". Also, we will use the terminology clients and copies interchangeably later in the analysis.

We can keep track of how each copy splits in each $\mathcal{D}_p$ for $p = 1, \ldots, L$. In this way, for each copy in $\mathcal{D}_L$, we can know which copy it belongs to in any $\mathcal{D}_p$ for each $p = 1, \ldots, L$. For simplicity, we will split each copy j in $\mathcal{D}_p$ into multiple copies $j_1, \ldots, j_r$ in $\mathcal{D}_L$ which all belong to it. These $j_1, \ldots, j_r$ will keep their own probability mass in $\mathcal{D}_p$, but their connection types (A, DC, IC) and $\tilde{\alpha}$-values (if not of type DC) will be inherited from j in $\mathcal{D}_p$. After processing this splitting, we can view $\mathcal{D}_1, \ldots, \mathcal{D}_L$ as the history of how each copy in $\mathcal{D}_L$ evolves from phase 1 to phase L. Further, the split process in the definition of "valid collection of copies" (Definition 8.5) can be regarded as turning some copies in each $\mathcal{D}_p$ from DC to IC.

One more issue to address is that in the collections of copies, we only maintain α-values for active or indirectly connected copies. We may lose track of the α-values of directly connected copies. To handle this issue, we define the α-value for each copy using the following procedure. We remark that this procedure is only for analysis purposes. The sequence of sequences $\mathcal{H}$ is enough for us to recover a solution. Hence, we do not need to pay attention to the time complexity of this procedure.

For RecoverAlpha, we have the following remarks:

- In the definition of the process, we will view free facilities and regular facilities equivalently (especially when we run RetrieveNewCopies), but in the later analysis of dual payment and Theorem 8.7, we still distinguish between them.
- The steps I) and II) in Stage 1 are well-defined since H_p is η-valid with respect to $\theta, (H_1, \ldots, H_{p-1}), \mathcal{D}_{p-1}$ (Definition 8.4).

Algorithm 3 (RecoverAlpha)**.**

Initialization: Set $\theta = 1$, $S = \emptyset$, $A = \mathcal{D}_L$, $DC = \emptyset$, $IC = \emptyset$, and $\alpha_j = \theta, R_j = 0$ for every $j \in \mathcal{D}_L$.

For each $p \in \{1, \ldots, L\}$:

- Let $\theta = (1+\varepsilon^3)^{p-1}$.
- **Stage 1.** For each facility h_t in the sequence H_p, perform the following steps:
  - I) If h_t is free, add h_t to S and jump to step IV). Otherwise, find a super-sequence $\mathcal{H}' \supseteq \langle H_1, \ldots, H_{p-1}\rangle \oplus (\{h_1, \ldots, h_{t-1}\})$ such that h_t is η-openable with respect to $\mathcal{H}'$. Let A' be the set of active clients induced by $\mathcal{D}_{new} = \text{RetrieveNewCopies}(\mathcal{D}_{p-1}, \mathcal{H}' - \langle H_1, \ldots, H_{p-1}\rangle)$.
  - II) Compute $(\tau_j)_{j\in A'}$ so that h_t is η-openable with $(\tau_j)_{j\in A'}$.
  - III) Add h_t to S. Set $\alpha_j \leftarrow \tau_j$ for every $j \in A' \cap B(h_t, \sqrt{\varepsilon\theta})$.
  - IV) For every $j \in A \cup IC$:
    - If $\alpha_j \geq \rho(1-\delta)d^2(j, h_t)$, move j to DC, and set R_j as follows:
      - if h_t is regular and j is directly connected in $\mathcal{D}_{new}$, set $R_j = \sqrt{\alpha_j/\rho}$;
      - otherwise, set $R_j = \sqrt{1-\delta}\cdot d(j, h_t)$.
    - Otherwise, if $j \in A$ and $\alpha_j \geq (1-\delta)d^2(j, h_t)$, move j to IC.
- **Stage 2.** Perform the following step and move to the next phase:
  - i) For every $j \in IC$, set $\alpha_j \leftarrow (1-\delta)d^2(j, S)$.
  - ii) For every $j \in A$, set $\alpha_j \leftarrow \min\{(1+\varepsilon^3)\theta, (1-\delta)d^2(j, S)\}$, and move it to IC when $\alpha_j = (1-\delta)d^2(j, S)$.
  - iii) For every $j \in DC$, if j's type in $\mathcal{D}_p$ is indirectly connected, move j to IC and set $\alpha_j \leftarrow (1-\delta)d^2(j, S)$, and $R_j \leftarrow 0$.
  - iv) Update $\theta \leftarrow (1+\varepsilon^3)\theta$.

- In Step IV) in Stage 1, we have two different ways to define R_j variables for newly directly connected clients. The reason we need different definitions is that h_t is only openable with respect to a super-sequence. If a client j is viewed as directly connected w.r.t. the super-sequence, we define R_j maximally, by enforcing $R_j^2 = \alpha_j/\rho$, because the $(\rho-1)\alpha_j/\rho$ part of α_j will no longer be used for opening other facilities (see, the proof Lemma 8.15). Otherwise, we define R_j as usual, by $R_j = \sqrt{1-\delta}d(j, h_t)$. This is a new feature for our analysis, compared with [CCG+26].
- We can show that RecoverAlpha produces the same solution as $(\mathcal{H}, \mathcal{DH})$ by comparing RecoverAlpha with RetrieveNewCopies, CompleteCopies and the definition of valid collection of copies (Definition 8.5):
  - In Stage 1, step III) of RecoverAlpha opens facilities in H_p one by one; and then step IV) of RecoverAlpha updates the α-values and connection types of all

clients in exactly the same way as RetrieveNewCopies.

– In Stage 2, steps i) and ii) of RecoverAlpha update the α-values and connection types of all clients in exactly the same way as the second step of CompleteCopies. Step iii) of RecoverAlpha moves some directly connected clients to indirectly connected, which is exactly the same as the split process in Definition 8.5 under our viewpoint. We shall remark that all copies moved from DC to IC are those in $IC \cup A$ in $\mathcal{D}_{p-1}$ because of our algorithm in Subsection 8.3.4.

Throughout the analysis, we let α_j^* be the final α-value of client $j \in \mathcal{D}_L$. With this RecoverAlpha procedure, we can prove two key lemmas, Lemmas 8.15 and 8.16, that are analogs of Lemmas 7.3 and 7.4, respectively. The proofs of them are almost identical to that for LogAdaptiveAlgorithm. There are only three minor changes:

1. Definition 8.2 of η-openability allows for a facility to be paid up to an $8n\eta$-additive difference and ensures every facility not to be overbid up to an $8n\eta$-additive difference. The first difference only appears in Lemma 8.15 below, while the second difference only appears in Lemma 8.16. These differences reflect in step I) and II) of stage 1.

2. Definition 8.4 of an η-valid sequence allows h_t is η-openable with respect to a super sequence $\mathcal{H}' \supseteq \mathcal{H} \oplus (\{h_1, \ldots, h_{t-1}\})$. This difference appears in the proof of Lemma 8.15, and (the last part of) the proof of Lemma 8.16, an analog of Lemma 7.4. This difference reflects in step I), II) and IV) of stage 1. In particular, a significant difference is in the proof of $r_{jt} \le R_j$ (Lemma 8.22).

3. Definition 8.5 of valid collections of copies allows for some directly connected copies to be moved to indirectly connected in Stage 2 of each phase. This difference only appears in (the last part of) the proof of Lemma 8.15 and the (first part of) the proof of Lemma 8.16. This difference reflects in step iii) of stage 2.

The combination of these two lemmas will imply Theorem 8.7.

Proof of Theorem 8.7. For each original client $x \in D$, let

$$D_x = \{j \in \mathcal{D}_L : oc(j) = x\}.$$

E: I modified this proof quite a bit. This needs to be checked.

RG: GPT finds no bugs for now.

Define

$$\alpha_x := \sum_{j \in D_x} w(j) \cdot \left(\alpha_j^* - (\rho - 1)R_j^2\right).$$

Since all copies of x are located at the same point as x, they have the same distance to every facility as x does. Also,

$$\sum_{j \in D_x} w(j) = 1.$$

Let

$$\Gamma' := \Gamma + 201\sqrt{\varepsilon}, \qquad \Lambda := \Gamma' + 20n\eta,$$

and run the robust algorithm with

$$\hat{f} := \Lambda f + 12n\eta.$$

Equivalently,

$$\hat{f} - 12n\eta = \Lambda f.$$

By Lemma 8.15, we have

$$\begin{aligned}\sum_{x\in D} d^2(x,S) &= \sum_{j\in\mathcal{D}_L} w(j)d^2(j,S)\\ &\le \frac{\sum_{j\in\mathcal{D}_L} w(j)\big(\alpha^*_j - (\rho-1)R_j^2\big) - |S_{\mathsf{reg}}|(\hat{f}-12n\eta)}{1-\delta}\\ &= \frac{\sum_{x\in D}\alpha_x - |S_{\mathsf{reg}}|\Lambda f}{1-\delta}. \end{aligned} \tag{40}$$

We next prove that α/Λ is feasible for the dual of the facility location LP with opening cost f. Fix any regular facility i. By Lemma 8.16, convexity of $[z]^+ = \max\{z,0\}$, and $\sum_{j\in D_x} w(j) = 1$, we get

$$\begin{aligned}\sum_{x\in D}[\alpha_x - \Gamma' d^2(i,x)]^+ &= \sum_{x\in D}\left[\sum_{j\in D_x} w(j)\big(\alpha^*_j - (\rho-1)R_j^2\big) - \Gamma' d^2(i,x)\right]^+\\ &= \sum_{x\in D}\left[\sum_{j\in D_x} w(j)\big(\alpha^*_j - (\rho-1)R_j^2 - \Gamma' d^2(i,j)\big)\right]^+\\ &\le \sum_{x\in D}\sum_{j\in D_x} w(j)\left[\alpha^*_j - (\rho-1)R_j^2 - \Gamma' d^2(i,j)\right]^+\\ &\le \hat{f} + 8n\eta. \end{aligned} \tag{41}$$

Let

$$D^* := \{x\in D : \alpha_x > \Lambda d^2(i,x)\}.$$

If $D^* = \emptyset$, then

$$\sum_{x\in D}[\alpha_x - \Lambda d^2(i,x)]^+ = 0 \le \Lambda f.$$

Otherwise,

$$\begin{aligned}\sum_{x\in D}[\alpha_x - \Lambda d^2(i,x)]^+ &= \sum_{x\in D^*}\big(\alpha_x - \Lambda d^2(i,x)\big)\\ &= \sum_{x\in D^*}\big(\alpha_x - \Gamma' d^2(i,x)\big) - 20n\eta\sum_{x\in D^*} d^2(i,x)\\ &\le \sum_{x\in D}[\alpha_x - \Gamma' d^2(i,x)]^+ - 20n\eta\sum_{x\in D^*} d^2(i,x)\\ &\le \hat{f} + 8n\eta - 20n\eta\sum_{x\in D^*} d^2(i,x). \end{aligned} \tag{42}$$

Every $x\in D^*$ satisfies $d^2(i,x)\ge 1$ (because of the assumptions we made in Lemmas 3.2 and 8.1 for the metric and Euclidean cases respectively). Since $D^*\neq\emptyset$, this implies

$$\sum_{x\in D^*} d^2(i,x) \ge 1.$$

Therefore, from Equation (42),

$$\sum_{x\in D}[\alpha_x - \Lambda d^2(i,x)]^+ \le \hat{f} + 8n\eta - 20n\eta = \hat{f} - 12n\eta = \Lambda f.$$

Thus, for every regular facility i,

$$\sum_{x \in D} \left[\frac{\alpha_x}{\Lambda} - d^2(i,x) \right]^+ \leq f.$$

Together with $\alpha_x \geq 0$, this shows that α/Λ is feasible for the dual of the facility location LP with opening cost f. Hence,

$$\sum_{x \in D} \frac{\alpha_x}{\Lambda} \leq opt_{\mathrm{LP}}(f).$$

Plugging this into Equation (40), we obtain

$$\begin{aligned}
\sum_{x \in D} d^2(x,S) &\leq \frac{\Lambda}{1-\delta} \left(\sum_{x \in D} \frac{\alpha_x}{\Lambda} - |S_{\mathsf{reg}}| f \right) \\
&\leq \frac{\Lambda}{1-\delta} \left(opt_{\mathrm{LP}}(f) - |S_{\mathsf{reg}}| f \right) \\
&= \frac{\Gamma + 201\sqrt{\varepsilon} + 20n\eta}{1-\delta} \left(opt_{\mathrm{LP}}(f) - |S_{\mathsf{reg}}| f \right).
\end{aligned}$$

This completes the proof. □

Note the weights satisfy $\sum_{j \in \mathcal{D}_L} w(j) = n$ because each original client has total weight 1 over all its copies. In the rest of this subsection, we focus on proving Lemmas 8.15 and 8.16. Because it is clear in the context we will only consider the copies as the clients, we will assume $D = \mathcal{D}_L$ in the proof and omit the "original clients" for simplicity. We start by showing the dual payment guarantee with respect to α^*, and then we prove that $(\alpha_j^* - (\rho-1)R_j^2)_{j \in D}$ is a feasible solution to the dual after appropriate scaling.

8.4.1 Dual Payment

Lemma 8.15. *We have* $\sum_{j \in \mathcal{D}_L} w(j) \cdot \left[(1-\delta) d^2(j,S) + (\rho-1) R_j^2 \right] + \sum_{i \in S_{\mathsf{reg}}} (\hat{f} - 12n\eta) \leq \sum_{j \in \mathcal{D}_L} w(j) \cdot \alpha_j^*$.

Proof. First, let us show that it suffices to prove that at the end of RecoverAlpha, we have

$$\sum_{j \in DC} w(j) \cdot \alpha_j \geq \sum_{j \in DC} w(j) \cdot \left[(1-\delta) d^2(j,S) + (\rho-1) R_j^2 \right] + \sum_{i \in S_{\mathsf{reg}}} (\hat{f} - 12n\eta). \tag{43}$$

Indeed, at the end of RecoverAlpha, $A = \emptyset$. For every $j \in IC$, we have $\alpha_j = (1-\delta) d^2(j,S)$ by the construction of the update steps in Stage 2. Because $R_j = 0$ for $j \in IC$, their contribution naturally satisfies the required bound. Therefore, Equation (43) implies the lemma.

To this end, we prove by induction on p that after the p-th phase in RecoverAlpha, Equation (43) holds. The base case is $p = 0$, i.e., at the beginning of the RecoverAlpha, where $DC, S = \emptyset$. Next, we show that this invariant is preserved after each phase of the algorithm.

The first case is that we only open free facilities in Stage 1 of the p-th phase. In this case, the term $\sum_{i \in S_{\mathsf{reg}}} (\hat{f} - 12n\eta)$ does not change. Suppose DC' are the clients that are moved from $IC \cup A$ to DC after stage 2 of this phase. In an earlier remark, we have argued that there is no copy moved from DC to $IC \cup A$. Consider any $j \in DC'$. Due to step IV) of Stage

1, we have $\alpha_j \geq \rho(1-\delta)d^2(j,h)$, where $h \in S$ is the first open facility satisfying the inequality. Furthermore, for this newly directly connected client, $R_j = \sqrt{1-\delta}d(j,h)$. As established in Section 7 (see the proof of Lemma 7.3), we have $(1-\delta)d^2(j,S)+(\rho-1)R_j^2 \leq \rho(1-\delta)d^2(j,h) \leq \alpha_j$. Thus, j's contribution to the LHS of Equation (43) is no less than that to the RHS. For other clients in $DC \setminus DC'$, their contributions to the LHS do not change while the RHS may only decrease because S only grows. Therefore, the inequality still holds.

The second case is that we open at least one regular facility i in Stage 1 of the p-th phase. The rest of the proof will focus on this case. Let us consider each step of stage 1, where we open a facility i, i.e., adding it to S. Suppose that $(\alpha, S, A, IC, DC, \theta)$ denotes the state right before opening i. Suppose that (S', A', IC', DC') denote the set of open facility in the super-sequence $\mathcal{H}'$ considered in the step i is opened, the set of active clients with respect to $\mathcal{D}_{new}$ (as in Step I) of the process, defined in terms of $\mathcal{H}'$), the set of indirectly connected clients with respect to $\mathcal{D}_{new}$ and the set of directly connected clients with respect to $\mathcal{D}_{new}$. Let $(\tau_j)_{j\in A'}$ be the τ-variables when opening i. Because $\mathcal{H}'$ is a super sequence of open facilities, we have $S' \supseteq S$, $DC' \supseteq DC$ and $A' \subseteq A$.

According to the third bullet of η-openability (Definition 8.2), we have

$$\hat{f} - 8n\eta \leq \sum_{j\in D} w(j) \cdot \texttt{bid}(j) \ ,$$

where

$$\texttt{bid}(j) = \begin{cases} [\tau_j - \rho(1-\delta)d^2(j,i)]^+ & \text{if } j \in A'; \\ [\alpha_j - \rho(1-\delta)d^2(j,i)]^+ & \text{if } j \in IC'; \\ (1-\delta)[d^2(j,S') - d^2(j,i)]^+ & \text{if } j \in DC'. \end{cases}$$

Next, we have a case-by-case discussion on the client's contribution to both sides of Equation (43) for each $j \in \mathcal{D}_L$. Our goal is to show that the contribution to the LHS minus the contribution to the RHS is at least $w(j) \cdot \texttt{bid}(j)$. Because of the third bullet of η-openability, we have the total contribution to the LHS is greater than that to the RHS by an additive factor of $4n\eta$. Throughout the discussion, for simplicity, we can ignore the $w(j)$ coefficient, because we have such a coefficient on their contributions to both sides.

- If $j \in A'$, this means $j \in A$.
  - If j is not DC after opening i, the contributions to both sides are 0 and we have $\tau_j < \rho(1-\delta)d^2(j,i)$, and thus $\texttt{bid}(j) = [\tau_j - \rho(1-\delta)d^2(j,i)]^+ = 0$.
  - If j is DC after opening i, the contributions to both sides are τ_j and $(1-\delta)d^2(j,i)+(\rho-1)R_j^2$. Because $j \notin DC'$, we have $R_j = \sqrt{1-\delta}d(j,i)$ and thus the difference is $\tau_j - \rho(1-\delta)d^2(j,i)$. Because j becomes DC after opening i, this quantity is greater than 0. We have the difference equal to $\texttt{bid}(j)$.
- If $j \in IC'$, this means $j \in A \cup IC$.
  - If j is not DC after opening i, the contributions to both sides are 0 and we have $\alpha_j < \rho(1-\delta)d^2(j,i)$, and thus $\texttt{bid}(j) = [\alpha_j - \rho(1-\delta)d^2(j,i)]^+ = 0$.
  - If j is DC after opening i, the contributions to both sides are α_j and $(1-\delta)d^2(j,i)+(\rho-1)R_j^2$. Because $j \notin DC'$, we have $R_j = \sqrt{1-\delta}d(j,i)$ and thus the difference is $\alpha_j - \rho(1-\delta)d^2(j,i)$. Because j becomes DC after opening i, this quantity is greater than 0. We have the difference equal to $\texttt{bid}(j)$.

- If $j \in DC$, this means $j \in DC'$. Then, the contribution of it to the LHS is 0. Its contribution to the RHS is $(1-\delta)\cdot(d^2(j, S\cup\{i\}) - d^2(j,S))$ because R_j is not changed. Further, we have $d^2(j,S\cup\{i\}) - d^2(j,S) = -[d^2(j,S) - d^2(j,i)]^+ \le -\texttt{bid}(j)/(1-\delta)$ because $S \subseteq S'$ and $d^2(j,S) \ge d^2(j,S')$. Therefore, the difference is no less than $\texttt{bid}(j)$.
- If $j \in DC' \setminus DC$, we discuss two cases.
  - If j is not moved to DC after opening i, the contributions to both sides are 0. Note that $\alpha_j \ge \rho(1-\delta)d^2(j,S')$ because $j \in DC'$. However, as $\alpha_j < \rho(1-\delta)d^2(j, S\cup\{i\}) \le \rho(1-\delta)d^2(j,i)$, we have $d^2(j,S') < d^2(j,i)$. This implies that $\texttt{bid}(j) = 0$. Therefore, the difference equals to $\texttt{bid}(j)$.
  - If j is moved to DC after opening i, we set $R_j = \sqrt{\alpha_j/\rho}$ at this time. Its contribution to the LHS is α_j. Its contribution to the RHS is $(1-\delta)d^2(j,S\cup\{i\}) + \frac{\rho-1}{\rho}\alpha_j$. The difference is then $\alpha_j/\rho - (1-\delta)d^2(j,S\cup\{i\})$. Note that because $j \notin DC$ but is directly connected after opening i, $d^2(j,S\cup\{i\}) = d^2(j,i)$. The difference further equals to $\alpha_j/\rho - (1-\delta)d^2(j,i)$, which is greater than 0. Note that $\alpha/\rho \ge (1-\delta)d^2(j,S')$ because $j \in DC'$, we have the difference $\ge (1-\delta)(d^2(j,S') - d^2(j,i))$. Combining these two lower bounds on the difference, we conclude that the difference is lower bounded by $\texttt{bid}(j)$.

Then, in stage 2, we only move some clients from DC to IC (step iii)). For each such client j, its contribution to the LHS of Equation (43) drops by α_j while its contribution to the RHS drops by $(1-\delta)d^2(j,S) + (\rho-1)R_j^2$. Note that our definition ensures that $R_j^2 \ge \alpha/\rho \ge (1-\delta)d^2(j,S)$ or $R_j^2 = (1-\delta)d^2(j,i) \ge (1-\delta)d^2(j,S)$ for some $i \in S$. According to Definition 8.5, for each such client j moved from DC to IC, we are guaranteed that $\alpha_j < \rho(1-\delta)d^2(j,S) + 4\eta \le (1-\delta)d^2(j,S) + (\rho-1)R_j^2 + 4\eta$. Therefore, the decrease of the LHS is at most the decrease of the RHS plus an additive 4η per client. Over all such clients, the total extra drop is at most $4\eta\sum w(j) = 4n\eta$. Since we accumulated at least $4n\eta$ surplus in stage 1, the overall invariant Equation (43) still safely holds. This completes the induction and thus the proof. □

8.4.2 Dual Feasibility

We prove Lemma 8.16 for dual feasibility. Throughout this part, we will assume that $\Gamma > \rho + 100\sqrt{\varepsilon}$.

Lemma 8.16. *For each regular facility i, we have*

$$\sum_{j\in\mathcal{D}_L} w(j)\cdot[\alpha_j^* - (\rho-1)R_j^2 - (\Gamma + 201\sqrt{\varepsilon})\cdot d^2(i,j)]^+ \le \hat{f} + 8n\eta,$$

where

- *if d is a (general) metric, we choose $\rho = 2.5, \Gamma = 4.9$; or*
- *if d is a Euclidean metric, we choose $\rho = 2, \Gamma = 3 + \ln(2)$.*

The following claim is an analog of Claim 7.5.

Claim 8.17 (No Over-Bidding)**.** *At any point in the algorithm, for every facility $i_0 \in \mathcal{F}$, we have*

$$\sum_{j\in DC} w(j)\cdot[(1-\delta)d^2(j,S) - d^2(j,i_0)]^+ + \sum_{j\in A\cup IC} w(j)\cdot[\alpha_j - \rho d^2(j,i_0)]^+ \le \hat{f}.$$

Proof. **Stage 1:** Consider an arbitrary facility $i_0 \in \mathcal{F}$. Suppose that a facility i' is opened with dual increases $(\tau_j)_{j\in A}$. The opening of i' has three potential effects on the clients: (1) it may strictly increase the α-values of active clients $j \in A \cap B(i', \sqrt{\varepsilon\theta})$ to τ_j and subsequently make them directly connected (DC), (2) it may make some existing active or indirectly connected (IC) clients directly connected to i', and (3) it may make some active clients indirectly connected.

Let us analyze how the left-hand side of our constraint is impacted by these changes. First, observe that a client j that was already directly connected before the opening of i' can only lower its contribution, since the distance $d^2(j, S)$ can only decrease when i' is added to S.

For a client j that was active or indirectly connected and now becomes a direct connect, its contribution changes from$w(j) \cdot [\alpha_j - \rho d^2(j, i_0)]^+$ to $w(j) \cdot [(1-\delta)d^2(j, S \cup \{i'\}) - d^2(j, i_0)]^+$.

If j is active and $\tau_j > \alpha_j = \theta$, this means that j is directly connected to facility i', which has just been opened in the step, such that $d^2(j, i') \le \varepsilon\theta$. This implies j will be directly connected to i' and $(1-\delta)\rho d^2(j, S \cup \{i'\}) = (1-\delta)\rho d^2(j, i') \le \alpha_j$ because ε is sufficiently small. Hence, we have

$$[(1-\delta)d^2(j, S \cup \{i'\}) - d^2(j, i_0)]^+ \le [(1-\delta)\rho d^2(j, S \cup \{i'\}) - \rho d^2(j, i_0)]^+ \le [\alpha_j - \rho d^2(j, i_0)]^+.$$

Next, we consider the case j is active or indirectly connected but $\tau_j = \alpha_j$. Because the client became directly connected, its dual value must have dominated the connection cost, meaning$\alpha_j \ge \rho(1-\delta)d^2(j, i')$. Since $\rho > 1$ and $[x]^+ \ge 0$, we have:

$$\begin{aligned}[(1-\delta)d^2(j, S \cup \{i'\}) - d^2(j, i_0)]^+ &\le \left[\frac{\tau_j}{\rho} - d^2(j, i_0)\right]^+ \\ &\le \left[\tau_j - \rho d^2(j, i_0)\right]^+.\end{aligned}$$

Thus, its contribution does not increase.

With the above discussions, we prove that the invariant holds throughout Stage 1.

Stage 2: In the second stage, we have three steps: decreasing the α-values of clients in IC to $(1-\delta)d^2(j, S)$; increasing the α-values of clients in A to $\min\{(1+\varepsilon^3)\theta, (1-\delta)d^2(j, S)\}$; and moving some clients from DC to IC.

The first step can only decrease the left-hand side of the claim. The third step changes the contribution of a moved client from $[(1-\delta)d^2(j, S) - d^2(j, i_0)]^+$ to $[\alpha_j - \rho d^2(j, i_0)]^+ = [(1-\delta)d^2(j, S) - \rho d^2(j, i_0)]^+$. Since $\rho > 1$, this also strictly decreases its contribution. Hence, it suffices to show that the continuous increase in the second step does not violate the claim.

Assume towards contradiction that at the end of the phase, if we increase α_j of every $j \in A$ to $\min\{(1+\varepsilon^3)\theta, (1-\delta)d^2(j, S)\}$, the claim is violated for a non-empty subset of facilities F'.

We select a "minimal" such counter-example: let $\tau' \le (1+\varepsilon^3)\theta$ be the smallest value such that if we set $\alpha'_j := \min\{\tau', (1-\delta)d^2(j, S)\}$, it satisfies

$$\begin{aligned}\sum_{j\in A} w(j)[\alpha'_j - \rho d^2(j, i)]^+ + \sum_{j\in IC} w(j)[\alpha_j - \rho d^2(j, i)]^+ \\ + \sum_{j\in DC} w(j)[(1-\delta)d^2(j, S) - d^2(j, i)]^+ = \hat{f}\end{aligned}$$

for some $i \in F'$. We remark that $\tau' \ge \theta$ since the constraint was strictly satisfied at the end of Phase 1. Furthermore, $i \notin S$ is not yet open; if it were open, there would be no active client $j \in A$ for which $\alpha'_j - \rho d^2(j, i)$ is strictly positive (since $\alpha'_j \le (1-\delta)d^2(j, S) \le (1-\delta)d^2(j, i) <$

$\rho d^2(j,i)$). Hence, the increase in α-values cannot cause i to violate the constraint if it were already opened, which contradicts $i \in F'$.

We now show that this facility i, combined with dual values $\tau_j = \alpha'_j$ for $j \in A \cap B(i, \sqrt{\varepsilon\theta})$ and $\tau_j = \alpha_j$ for $j \in A \setminus B(i, \sqrt{\varepsilon\theta})$, satisfies the conditions of η-openability. In other words, i is an openable facility that was not yet opened, which contradicts the proper completion of the first stage's while-loop.

We verify the conditions of η-openability one-by-one. The first two bullets are directly satisfied by our choice of $\tau_j \le \tau'$ and the definition of α'_j. For the third bullet (facility i is paid for):

$$\begin{aligned}
&\sum_{j\in A} w(j)\cdot[\tau_j - \rho(1-\delta)d^2(j,i)]^+ + \sum_{j\in IC} w(j)\cdot[\alpha_j - \rho(1-\delta)d^2(j,i)]^+ \\
&\quad + \sum_{j\in DC} w(j)\cdot[(1-\delta)d^2(j,S) - (1-\delta)d^2(j,i)]^+ \\
&= \sum_{j\in A\cap B(i,\sqrt{\varepsilon\theta})} w(j)\cdot[\alpha'_j - \rho(1-\delta)d^2(j,i)]^+ + \sum_{j\in A\setminus B(i,\sqrt{\varepsilon\theta})} w(j)\cdot[\alpha_j - \rho(1-\delta)d^2(j,i)]^+ \\
&\quad + \sum_{j\in IC} w(j)\cdot[\alpha_j - \rho(1-\delta)d^2(j,i)]^+ + \sum_{j\in DC} w(j)\cdot[(1-\delta)d^2(j,S) - (1-\delta)d^2(j,i)]^+ \\
&\ge \sum_{j\in A\cap B(i,\sqrt{\varepsilon\theta})} w(j)\cdot[\alpha'_j - \rho d^2(j,i)]^+ + \sum_{j\in A\setminus B(i,\sqrt{\varepsilon\theta})} w(j)\cdot[\alpha'_j - \rho d^2(j,i)]^+ \\
&\quad + \sum_{j\in IC} w(j)\cdot[\alpha_j - \rho d^2(j,i)]^+ + \sum_{j\in DC} w(j)\cdot[(1-\delta)d^2(j,S) - d^2(j,i)]^+ \\
&= \hat{f}.
\end{aligned}$$

Here, for the subset outside the ball $A \setminus B(i, \sqrt{\varepsilon\theta})$, we used the fact that $\alpha_j - \rho(1-\delta)d^2(j,i) \ge (1+\varepsilon^2)\alpha_j - \rho d^2(j,i) \ge \alpha'_j - \rho d^2(j,i)$ when $d^2(j,i) \ge \varepsilon\theta$ (recalling that $\alpha_j = \theta$ for all active clients).

It remains to verify the fourth bullet of η-openability (dual feasibility). Suppose toward contradiction that there is a facility i_0 such that

$$\sum_{j\in DC} w(j)[(1-\delta)d^2(j,S) - d^2(j,i_0)]^+ + \sum_{j\in IC} w(j)[\alpha_j - \rho d^2(j,i_0)]^+ + \sum_{j\in A} w(j)[\tau_j - \rho d^2(j,i_0)]^+ > \hat{f}.$$

Because $\tau_j \le \alpha'_j$ for all $j \in A$, this implies:

$$\begin{aligned}
&\sum_{j\in DC} w(j)[(1-\delta)d^2(j,S) - d^2(j,i_0)]^+ + \sum_{j\in IC} w(j)[\alpha_j - \rho d^2(j,i_0)]^+ + \sum_{j\in A} w(j)[\alpha'_j - \rho d^2(j,i_0)]^+ \\
&\ge \sum_{j\in DC} w(j)[(1-\delta)d^2(j,S) - d^2(j,i_0)]^+ + \sum_{j\in IC} w(j)[\alpha_j - \rho d^2(j,i_0)]^+ + \sum_{j\in A} w(j)[\tau_j - \rho d^2(j,i_0)]^+ \\
&> \hat{f}.
\end{aligned}$$

which directly contradicts the minimality of τ'. Thus, i is openable, representing a contradiction. □

Next, we fix an arbitrary regular facility $i \in \mathcal{F}$ and show that the dual-feasibility for i is satisfied at the end of the robust algorithm. We say that facility i is frozen if there exists some open facility i_0 such that

$$d(i,i_0) \le 2\cdot\sqrt{\varepsilon\theta}\ .$$

Modified α-values. For analysis purposes, we introduce a modification of the α-values for each client (copy) $j \in \mathcal{D}_L$ in terms of the fixed facility i. We denote this modified α-value as α_j^c. This modified α^c-value will be used to write the analog of the dual constraints.

The modifications happen only when we open a facility i_0 that does not freeze i, i.e., when $d(i, i_0) > 2\sqrt{\varepsilon\theta}$. More specifically, at the stage with time θ, if we open such a facility i_0 that does not freeze i, we keep α_j^c as the value before opening i_0. That is, we do not increase α_j^c to τ_j if $j \in B(i_0, \sqrt{\varepsilon\theta})$ (step 1b) for such i_0. In other steps (step 1b for a facility i_0 that freezes i or step 2), we still modify α_j^c in the same way we modify α_j.

In the rest of this subsection, we will consider α^c instead of α^* in writing the constraints. To ensure that the α^*-value can ensure the dual feasibility, we first show that α_j^c, α_j^* are close to each other for every client $j \in \mathcal{D}_L$.

The following lemma and its proof are identical to Lemma 7.6.

Lemma 8.18. *For any client $j \in \mathcal{D}_L$, we have $\alpha_j^* - \alpha_j^c \le \varepsilon^2 \cdot d^2(j, i)$.*

Now, we assume that

$$D^* = \{j \in \mathcal{D}_L : \alpha_j^c - (\rho - 1)R_j^2 > \Gamma d^2(j, i)\} \ . \tag{44}$$

Then, it suffices to show that

$$\sum_{j \in D^*} w(j) \cdot (\alpha_j^c - (\rho - 1)R_j^2) \le \hat{f} + 8n\eta + (\Gamma + 200\sqrt{\varepsilon}) \cdot \sum_{j \in D^*} w(j) \cdot d^2(j, i) \ .$$

Establishing the constraints. Up to this point, our arguments closely mirror the dual feasibility proof for the log-adaptive algorithm in Section 7. However, the remaining part of the proof needs some changes from Section 7, due to the fact that a facility may be η-openable with respect to a super-sequence instead of the sequence that the algorithm actually executes.

Let i_0 be the facility that freezes i, which opens in (stage 1 of) phase p. Let $\theta = (1+\varepsilon^3)^{p-1}$ be the active dual parameter at the beginning of phase p. Let $\mathcal{H}$ be the sequence of opened facilities up to i_0, and let $\mathcal{H}' \supseteq \mathcal{H}$ be the super-sequence with respect to which i_0 is η-openable.

As in Section 7, we order the clients in D^* according to the order they are removed from A (in the algorithm), breaking ties in the increasing order of their α^c-values. The only additional rule is for the clients who are in A right before i_0 is opened in $\mathcal{H}$; let us call this set of active clients A. By Claim 8.20, they are the last several clients in the above order of D^*. Let $A^F \subseteq A$ be the clients in D^* who are still active right before i_0 is opened according to the collection of copies induced by RetrieveNewCopies$(\mathcal{D}_{p-1}, H'_p)$. Let $A' = (D^* \cap A) \setminus A^F$ be the clients that become inactive during the extra prefix in $\mathcal{H}'$. In our ordering for $D^* \cap A$, we place A' first and A^F after, sorting within each group in the increasing order of their α_j^c values.

Let a be the index of the last client in $D^* \setminus A^F$, meaning the clients in $[a+1:s]$ are exactly A^F. For each client $t \in [a]$, we define S^t, DC^t, and IC^t to be the sets of opened facilities, directly connected clients in $[t-1]$, and indirectly connected (or active) clients in $[t-1]$, evaluated at the beginning of the exact stage t becomes inactive. For each client $t \in [a+1:s]$, we define S^t, DC^t, and IC^t to be the sets of opened facilities, directly connected clients in $[t-1]$, and indirectly connected (or active) clients in $[t-1]$, evaluated right before t becomes inactive in $\mathcal{H}'$. Similarly, let S^F, IC^F, and DC^F be the set of opened facilities, indirectly connected, and directly connected clients right before i_0 is opened evaluated with respect to the super-sequence $\mathcal{H}'$.

We remark that the definitions of S^F, IC^F, DC^F, and A^F here depend on the super-sequence $\mathcal{H}'$, while in Section 7, they only depend on the sequence $\mathcal{H}$ executed by the algorithm.

To complete the proof, we will now apply the unified constraint lemmas (Lemmas 4.11, 4.12 and 5.20) evaluated over this super-sequence state.

The two following claims are analogs of Claims 7.7 and 7.8. The proof of Claim 8.19 is identical to that of 7.7. For Claim 8.20, we strengthen the argument to "client j becomes directly connected" upon "client j becomes active".

Claim 8.19. *For any $t \in D^*$ that becomes inactive strictly before i becomes frozen,*

$$\sum_{j \in DC^t} w(j)[(1-\delta)d^2(j,S^t) - d^2(j,i)]^+ + \sum_{j \in IC^t} w(j)[\alpha_j^c - \rho d^2(j,i)]^+ + \sum_{j>t} w(j)[\alpha_t^c - \rho d^2(j,i)]^+ \le \hat{f} + 8n\eta.$$

Claim 8.20. *If i is frozen by i_0 with $(\tau_j)_{j \in A}$, then for every $j \in D^* \cap A$, j becomes directly connected when opening i_0 and $\alpha_j^c \le \tau_j$.*

Proof. Applying the same proof of Claim 7.8, we can show that j is inactive. Next, we present a short argument for why it is directly connected for our choices of Γ.

Note that $\theta \le \alpha_j^c$ and $\alpha_j^c \ge \Gamma \cdot d^2(j,i)$ because $j \in D^* \cap A$. Note that $d(j,i_0) \le d(j,i) + d(i,i_0) \le d(j,i) + 2\sqrt{\varepsilon\theta}$. This implies that

$$\begin{aligned} d^2(j,i_0) \le d^2(j,i) + 4\varepsilon\theta + 4\sqrt{\varepsilon\theta}d(j,i) &\le \alpha_j^c/\Gamma + 4\varepsilon\theta + 4\sqrt{\varepsilon\theta}\sqrt{\alpha_j^c/\Gamma} \\ &\le \alpha_j^c/\Gamma + 4\varepsilon\alpha_j^c + 4\sqrt{\varepsilon/\Gamma}\alpha_j^c \\ &\le (1/\Gamma + 8\sqrt{\varepsilon})\alpha_j^c. \end{aligned}$$

Because $\Gamma > \rho + 100\sqrt{\varepsilon}$, $\rho \in [2,3]$ and ε is sufficiently small, we have $\alpha_j^c \ge \rho d^2(j,i_0)$. □

The following claim is the analog of Claim 7.9.

Claim 8.21. *For any $t \in [s]$,*

$$\sum_{j \in DC^t} w(j)((1-\delta)d^2(j,S^t) - d^2(j,i)) + \sum_{j \in IC^t} w(j)(\alpha_j^c - \rho d^2(j,i)) + \sum_{j \ge t} w(j)(\alpha_t^c - \rho d^2(j,i)) \le \hat{f} + 8n\eta.$$

Proof. Consider the point in the execution evaluated against the super-sequence $\mathcal{H}'$ at which t becomes inactive. As established by Claim 8.20, if a facility i_0 freezes i, every client $j \in D^*$ that is active right before i_0 is opened becomes inactive during the exact step i_0 is opened. Therefore, no client in D^* can become inactive *after* i_0 is opened in $\mathcal{H}'$. This partitions our ordered clients $t \in D^*$ into exactly two cases:

Case 1: $t \in [a]$. Client t becomes inactive strictly before i_0 is opened in $\mathcal{H}'$. Because every facility in the valid super-sequence $\mathcal{H}'$ is η-openable, the No Over-Bidding invariant (Claim 8.17) rigorously holds at every sequential step of its execution. We evaluate this depending on the stage in which t becomes inactive:

Stage 1: Suppose t becomes inactive due to the discrete opening of a facility $h_k \in \mathcal{H}'$. Because h_k precedes i_0 (the facility that freezes i), this event happens strictly before i is frozen. Therefore, h_k does not freeze i. By our definition of α^c, no client in D^* increases its α^c-value, meaning the modified dual α^c does not incorporate any Step 1b increases from h_k. Thus, α_t^c is exactly the active dual parameter θ right before h_k is opened. We evaluate the No Over-Bidding constraint at the state right before h_k opens:

$$\sum_{j \in DC^t} w(j)[(1-\delta)d^2(j,S^t) - d^2(j,i)]^+ + \sum_{j \in IC^t \cup A^t} w(j)[\alpha_j - \rho d^2(j,i)]^+ \le \hat{f} + 8n\eta.$$

For $j \in DC^t$, the term trivially bounds $(1-\delta)d^2(j,S^t) - d^2(j,i)$. For $j \in IC^t$, the modified dual α^c never exceeds the true dual prior to a freezing step, so $\alpha_j \geq \alpha^c_j$. For $j \geq t$, because our ordering places clients by the time they become inactive, they are still active in A^t at this moment, meaning their actual dual is $\alpha_j = \theta = \alpha^c_t$. Thus, $[\alpha_j - \rho d^2(j,i)]^+ \geq \alpha^c_t - \rho d^2(j,i)$.

Stage 2: Suppose t becomes inactive due to the continuous update of α-values in Stage 2. We evaluate the No Over-Bidding constraint at the exact moment t hits its threshold. At this moment, the modified dual exactly tracks the true dual, so $\alpha^c_t = \alpha_t$. For $j \in IC^t$, we again have $\alpha_j \geq \alpha^c_j$. For $j \geq t$, these clients either remain active or become inactive at the same moment but are sorted after t (meaning $\alpha^c_j \geq \alpha^c_t$). In both cases, their true dual at this moment satisfies $\alpha_j \geq \alpha_t = \alpha^c_t$. Thus, $[\alpha_j - \rho d^2(j,i)]^+ \geq \alpha^c_t - \rho d^2(j,i)$.

In both Stage 1 and Stage 2, the subsets DC^t, IC^t, and $\{j \in D^* : j \geq t\}$ are mutually disjoint. Restricting the summations to these subsets and dropping the non-negative $[\cdot]^+$ operators rigorously establishes the algebraic lower bound:

$$\sum_{j\in DC^t} w(j)((1-\delta)d^2(j,S^t)-d^2(j,i))+\sum_{j\in IC^t} w(j)(\alpha^c_j-\rho d^2(j,i))+\sum_{j\geq t} w(j)(\alpha^c_t-\rho d^2(j,i)) \leq \hat{f}+8n\eta.$$

Case 2: $t \in [a+1:s]$**.** Client t remains active right before i_0 is opened in $\mathcal{H}'$ (meaning $t \in A^F$) and becomes inactive exactly when i_0 is opened. Because all clients in A^F become inactive simultaneously at this step, and ties in our ordering are broken by increasing α^c-value, we are guaranteed that $\alpha^c_t \leq \alpha^c_j$ for all $j \geq t$. Furthermore, evaluated right before i_0 is opened in $\mathcal{H}'$, we identically have $S^t = S^F$, $DC^t \subseteq DC^F$, and $IC^t = IC^F \cup \{j \in A^F : j < t\}$.

Consider the fourth bullet of η-openability (dual feasibility) for i_0 evaluated at the fixed facility i, just before i_0 is opened in $\mathcal{H}'$:

$$\sum_{j\in DC^F} w(j)[(1-\delta)d^2(j,S^F)-d^2(j,i)]^+ + \sum_{j\in IC^F} w(j)[\alpha_j-\rho d^2(j,i)]^+ + \sum_{j\in A^F} w(j)[\tau_j-\rho d^2(j,i)]^+ \leq \hat{f}+8n\eta.$$

Since the $[\cdot]^+$ terms are strictly non-negative, restricting the summations to disjoint subsets of interest maintains the upper bound. We then drop the non-negative operators entirely to establish a strict algebraic lower bound:

- For $j \in DC^t \subseteq DC^F$: Because $S^t = S^F$, we have: $[(1-\delta)d^2(j,S^F) - d^2(j,i)]^+ \geq (1-\delta)d^2(j,S^t) - d^2(j,i)$.
- For $j \in IC^t \cap IC^F$: The modified dual never exceeds the true dual prior to freezing, so $\alpha_j \geq \alpha^c_j$. Thus: $[\alpha_j - \rho d^2(j,i)]^+ \geq \alpha_j - \rho d^2(j,i) \geq \alpha^c_j - \rho d^2(j,i)$.
- For $j \in IC^t \cap A^F$: These clients are in A^F but ordered before t. By Claim 8.20, we have $\tau_j \geq \alpha^c_j$. Evaluated in the summation over A^F, we have: $[\tau_j - \rho d^2(j,i)]^+ \geq \tau_j - \rho d^2(j,i) \geq \alpha^c_j - \rho d^2(j,i)$. (Combining this with the previous bullet perfectly reconstructs the $\alpha^c_j - \rho d^2(j,i)$ term for all $j \in IC^t$).
- For $j \geq t$: These clients are exactly $\{j \in A^F : j \geq t\}$. By Claim 8.20, $\tau_j \geq \alpha^c_j$. Applying our tie-breaking rule, $\alpha^c_j \geq \alpha^c_t$, meaning $\tau_j \geq \alpha^c_t$. Evaluated in the A^F sum, we have: $[\tau_j - \rho d^2(j,i)]^+ \geq \tau_j - \rho d^2(j,i) \geq \alpha^c_t - \rho d^2(j,i)$.

Because all evaluated subsets are disjoint and strictly contained within the original η-openability sums, summing these lower bounds yields:

$$\sum_{j\in DC^t} w(j)((1-\delta)d^2(j,S^t)-d^2(j,i))+\sum_{j\in IC^t} w(j)(\alpha^c_j-\rho d^2(j,i))+\sum_{j\geq t} w(j)(\alpha^c_t-\rho d^2(j,i)) \leq \hat{f}+8n\eta.$$

This matches the required inequality, completely resolving the case. □

For any client $t \in [s]$ and $j \in DC^t$, define

$$r_{jt} := \sqrt{1-\delta} d(j, S^t).$$

Next, we prove that $r_{jt} \leq R_j$ for every $t \in [s]$ and $j \in DC^t$.

Lemma 8.22. *For every $t \in [s]$ and $j \in DC^t$, we have $r_{jt} \leq R_j$.*

Proof. We discuss two cases.

The first case is if $t \in [a]$. In this case, if $j \in DC^t$, it means $j \in DC$ and $(r_{jt})^2 = (1-\delta)d^2(j, S)$ at the beginning of the stage t becomes inactive. Therefore, at the beginning of the stage, we have defined R_j as $R_j = \sqrt{1-\delta} d(j, h)$ for some $h \in S$ or $R_j = \sqrt{\alpha_j/\rho}$. In both case, we can derive $r_{jt} \leq R_j$ because $d^2(j, S) \leq d^2(j, h)$ and $(1-\delta)d^2(j, S) \leq \alpha_j/\rho$.

The second case is if $t > a$. Let S be the set of open facilities right before i is frozen. We have $S = S^t$ by our earlier definition for this case. If $j \in DC^t$, there are two subcases:

- Suppose that $j \in DC$ right before i is frozen. This means $(r_{jt})^2 = (1-\delta)d^2(j, S)$, and $R_j = \sqrt{1-\delta} d(j, h)$ or $R_j = \sqrt{\alpha_j/\rho}$. As discussed earlier in this proof, we have $r_{jt} \leq R_j$.
- Otherwise (corresponding to the case j is directly connected only when it is evaluated in the super sequence $\mathcal{H}'$), we have $R_j = \sqrt{\alpha_j/\rho}$ because j becomes directly connected immediately after i is frozen (Claim 8.20). We also have $(1-\delta)d^2(j, S^t) \leq \alpha_j/\rho$ because j is directly connected with respect to S^t. Therefore, we have $r_{jt} \leq R_j$ by the definition of r_{jt}.

A final remark is that in the proof, we are considering the final real α-values instead of the α^c-values. □

General metric

Now, we will show Lemma 8.16 for $\rho = 2.5$ and $\Gamma = 4.9$. To this end, we show that $(IC^t), (DC^t), (\alpha_j^c), (R_j), (r_{jt})$ satisfy all the constraints of Lemma 4.12 (with the right-hand side of the capacity constraint replaced by $\hat{f} + 8n\eta$).

The proof of this is strictly identical to the one in Section 7, with only two minor adaptations due to the super-sequence: First, the last constraint (the capacity bound) is now directly provided by Claim 8.21 with the updated right-hand side $\hat{f} + 8n\eta$. Second, we will use Lemma 8.22 to argue that $r_{jt} \leq R_j$ still holds.

The remaining bounds, including $\alpha_t^c \leq (r_{jt} + d(j, t))^2$, follow exactly the same triangle inequality logic as in the log-adaptive part because $\alpha_t^c \leq (1-\delta)d^2(t, S^t)$ remains valid for all t (for $t \in [a+1 : s]$, we have $\alpha_t^c \leq \tau_t \leq (1-\delta)d^2(t, S^F) = (1-\delta)d^2(t, S^t)$ by the second bullet of η-openability).

Hence, applying Lemma 4.12 with the adjusted right-hand side, we get:

$$\sum_{j \in [s]} w(j) \cdot (\alpha_j^c - (\rho - 1)R_j^2) \leq \hat{f} + 8n\eta + \Gamma \cdot \sum_{j \in [s]} w(j) \cdot d^2(j, i).$$

This completes the proof of Lemma 8.16 for $\rho = 2.5$ and $\Gamma = 4.9$.

Euclidean metric

Now, we will show Lemma 8.16 for $\rho = 2$, $\Gamma = 3 + \ln 2$, and d a Euclidean metric. Let a be the index of the last client that becomes inactive strictly before i_0 is opened in the super-sequence $\mathcal{H}'$.

The following claim is the analog of Claim 7.10.

Claim 8.23. *For each $t \in [a]$, $j \in DC^t$ and $\ell \in IC^t \cup \{t\}$, we have:*

$$\alpha_\ell^c \le (r_{jt} + d(j, \ell))^2.$$

Proof. We first prove the bound for $t \in [a]$. Consider the exact state in the execution evaluated at the beginning of the stage where t becomes inactive. By definition, S^t is the set of open facilities at this state.

Consider any client $\ell \in IC^t \cup \{t\}$. As $\ell \le t$, ℓ becomes inactive at this state or earlier. We prove that $\alpha_\ell^c \le (1-\delta)d^2(\ell, S^t)$ by discussing two cases:

- If ℓ became inactive strictly before the phase containing t's inactivation, then ℓ is indirectly connected at the beginning of this phase. Because of the Stage 2 continuous updates at the end of the previous phase, we have $\alpha_\ell^c \le \alpha_\ell = (1-\delta)d^2(\ell, S_{prev})$. Because the set of open facilities only grows, $S_{prev} \subseteq S^t$, meaning the distance can only decrease. Thus, $\alpha_\ell^c \le (1-\delta)d^2(\ell, S^t)$.
- Otherwise, ℓ becomes inactive during the same phase as t (this includes the case $\ell = t$). Let θ_t be the active dual parameter at the beginning of this phase. Since ℓ is active at this time, $\alpha_\ell^c \le \alpha_\ell = \theta_t$. We must strictly have $\theta_t \le (1-\delta)d^2(\ell, S^t)$, because if $\theta_t > (1-\delta)d^2(\ell, S^t)$, ℓ would have been moved to IC during the Stage 2 continuous update of the previous phase (when the threshold exceeded the connection cost), contradicting that ℓ is still active at the beginning of the current phase. Therefore, $\alpha_\ell^c \le \theta_t \le (1-\delta)d^2(\ell, S^t)$.

We have established that $\alpha_\ell^c \le (1-\delta)d^2(\ell, S^t)$. Because the triangle inequality holds for the Euclidean metric d, we strictly have $d(\ell, S^t) \le d(j, S^t) + d(j, \ell)$. Taking the square root of our established bound, we get:

$$\begin{aligned}\sqrt{\alpha_\ell^c} &\le \sqrt{1-\delta}\, d(\ell, S^t) \\ &\le \sqrt{1-\delta}(d(j, S^t) + d(j, \ell)) \\ &= r_{j,t} + \sqrt{1-\delta}\, d(j, \ell).\end{aligned}$$

Since $\delta > 0$, we have $\sqrt{1-\delta} < 1$. Because metric distances are strictly non-negative, replacing the scaled distance with the full distance preserves the inequality:

$$\sqrt{\alpha_\ell^c} \le r_{j,t} + d(j, \ell).$$

Squaring both sides yields $\alpha_\ell^c \le (r_{j,t} + d(j, \ell))^2$, completing the proof. □

We will make several cases according to how i gets frozen.

Case 1: All clients in D^* become inactive strictly before i is frozen. In this case, we have $a = s$. Let us show that $(IC^t), (DC^t), (\alpha_j^c), (R_j)$, and (r_{jt}) satisfy all the constraints of Lemma 4.11, where the right-hand side of the capacity bound is adjusted to $\hat{f} + 8n\eta$.

First, constraint (1) holds because of Lemma 8.22. Second, since $a = s$, Claim 8.23 implies that constraint (2) is strictly satisfied for all $t \in [s]$. Finally, Claim 8.21 directly establishes constraint (3) with the relaxed right-hand side $\hat{f} + 8n\eta$.

Hence, we can seamlessly apply Lemma 4.11 (replacing $\hat{f}$ with $\hat{f} + 8n\eta$) to obtain:

$$\sum_{j\in[s]} w(j) \cdot (\alpha_j^c - (\rho - 1)R_j^2) \le \hat{f} + 8n\eta + \Gamma \cdot \sum_{j\in[s]} w(j) \cdot d^2(j, i).$$

This perfectly matches our target inequality, completing the proof for this case.

Case 2: Some clients in D^* become inactive when i is frozen. In this case, we assume that i_0 freezes i in the super-sequence $\mathcal{H}'$. Let IC^F be the set of indirectly connected clients right before i_0 is opened in $\mathcal{H}'$, and let A^F be the set of active clients in D^* right before i_0 is opened (meaning $a < s$). We define:

$$L := \{j \in IC^F : d(j, i) > \sqrt{\varepsilon\theta}\}.$$

Note that IC^F is disjoint from A^F, meaning $j \notin IC^F$ for any $j \in [a + 1 : s]$.

We will further discuss two subcases to prove Lemma 8.16.

Subcase 2(a): $[a + 1 : s] \cap B(i_0, \sqrt{\varepsilon\theta}) = \emptyset$ or $\sum_{j\in L} w(j) > \varepsilon \cdot \sum_{j\in[a+1:s]} w(j)$. We redefine DC^s and IC^s as the respectively directly connected and active or indirectly connected client in $[s - 1]$ at the beginning of the stage s becomes inactive. First, let us bound the total increment of α_j^c when i_0 is opened.

The following claim is the analog of Claim 7.11.

Claim 8.24. *The total increment of α_j^c for $j \in D^*$ when i_0 is opened is at most $\varepsilon \sum_{j\in D^*} w(j)d^2(j, i)$.*

Proof. If $[a + 1 : s] \cap B(i_0, \sqrt{\varepsilon\theta}) = \emptyset$, or if no active client in $[a + 1 : s]$ strictly increases its α^c-value when i_0 is opened, the total increment for all $j \in D^*$ is exactly 0. Because weights and squared distances are non-negative, the bound $0 \le \varepsilon \sum_{j\in D^*} w(j)d^2(j, i)$ holds trivially.

Next, we consider the case where $\sum_{j\in L} w(j) > \varepsilon \cdot \sum_{j\in[a+1:s]} w(j)$.

Let $U = [a + 1 : s] = A^F$ denote the set of clients in D^* that become inactive exactly when i is frozen (i.e., when i_0 is opened in $\mathcal{H}'$). For any client $j \in [1 : a]$, j became inactive strictly before this step. By the algorithm's rules, the dual values of inactive clients are never increased during Stage 1. Therefore, the total increment of α_j^c across all $j \in D^*$ is exactly the sum of the increments for the active clients in U.

Right before i_0 is opened, every $j \in U$ is active, meaning its dual value is exactly $\alpha_j = \theta$. When i_0 is opened, its modified dual value α_j^c is bounded by τ_j. By the second bullet of η-openability (Definition 8.2), the maximum increased value is bounded by $\tau_j \le (1 + \varepsilon^3)\theta$. Thus, for every $j \in U$, the increment is bounded by $\varepsilon^3\theta$.

Summing this over all $j \in U$, the total increment for D^* is bounded by:

$$\sum_{j\in U} w(j)(\alpha_j^c - \alpha_j) \le \varepsilon^3\theta \sum_{j\in U} w(j).$$

By our Subcase 2(a) assumption, we have $\sum_{j\in L} w(j) > \varepsilon \sum_{j\in U} w(j)$, which algebraically rearranges to:

$$\sum_{j\in U} w(j) < \frac{1}{\varepsilon}\sum_{j\in L} w(j).$$

Substituting this weight bound into our total increment strictly yields:

$$\sum_{j\in U} w(j)(\alpha_j^c - \alpha_j) < \varepsilon^3\theta\left(\frac{1}{\varepsilon}\sum_{j\in L} w(j)\right) = \varepsilon^2\theta\sum_{j\in L} w(j).$$

Recall the definition of the subset $L = \{j \in IC^F : d(j,i) > \sqrt{\varepsilon\theta}\}$. For every client $j \in L$, we strictly have $d^2(j,i) > \varepsilon\theta$, which rearranges to $\theta < \frac{1}{\varepsilon}d^2(j,i)$. Substituting this geometric upper bound for θ into our increment expression, we obtain:

$$\sum_{j\in U} w(j)(\alpha_j^c - \alpha_j) < \varepsilon^2\sum_{j\in L} w(j)\left(\frac{d^2(j,i)}{\varepsilon}\right) = \varepsilon\sum_{j\in L} w(j)d^2(j,i).$$

Finally, note that $L \subseteq IC^F \subseteq [1:a] \subset D^*$. Because client weights $w(j)$ are strictly positive and squared metric distances $d^2(j,i)$ are non-negative, the sum over the subset L is trivially bounded by the sum over the entire set D^*:

$$\varepsilon\sum_{j\in L} w(j)d^2(j,i) \le \varepsilon\sum_{j\in D^*} w(j)d^2(j,i).$$

This matches the target inequality, completely resolving the claim. □

Applying Claim 8.17 at the beginning of the stage that s becomes inactive, this lemma implies the following bound:

$$w(s)(\alpha_s^c - \rho d^2(s,i)) + \sum_{\ell\in IC^s} w(\ell)(\alpha_\ell^c - \rho d^2(\ell,i)) + \sum_{j\in DC^s} w(j)(r_{js}^2 - d^2(j,i)) \le (\hat{f} + 8n\eta) + \varepsilon\sum_{j\in D^*} w(j)d^2(j,i).$$

Because we strictly redefined IC^s, DC^s, and S^s according to the sets IC, DC, S at the beginning of the stage that s becomes inactive, Claim 8.23 perfectly generalizes for $t = s$. Thus, we can cleanly apply Lemma 5.20, replacing $\hat{f}$ with $(\hat{f} + 8n\eta) + \varepsilon\sum_{j\in D^*} w(j)d^2(j,i)$ and using $L = \emptyset$ and $\varepsilon' = 0$ (the internal ε-parameter in the lemma). This directly gives:

$$\sum_{j\in[s]} w(j)\cdot(\alpha_j^c - (\rho-1)R_j^2) \le (\hat{f} + 8n\eta) + (\Gamma+\varepsilon)\cdot\sum_{j\in[s]} w(j)\cdot d^2(j,i).$$

This completes the proof for this subcase.

Subcase 2(b): $[a+1:s] \cap B(i_0, \sqrt{\varepsilon\theta}) \neq \emptyset$ and $\sum_{j\in L} w(j) \le \varepsilon\cdot\sum_{j\in[a+1:s]} w(j)$. In this case, we again define $DC^s = DC^F$ and $IC^s = IC^F \cup A^F$ as the sets of directly connected and active/indirectly connected clients respectively, right before s becomes inactive (i.e., right before i_0 is opened evaluated in the super-sequence $\mathcal{H}'$). Let $S^s = S^F$.

We show that a, L, $(IC^t), (DC^t), (\alpha_j^c), (R_j)$, and (r_{jt}) strictly satisfy all the constraints of Lemma 5.20, with the capacity constraint relaxed by $8n\eta$. Conditions (a), (b), (c), (1), (2),

and (3) of the lemma hold by the same arguments established for Case 1 and the structural definitions of the sets. It remains to rigorously prove that condition (4) is satisfied. Because $L \subseteq IC^F$, $IC^F \subseteq [a]$, and $IC^F \subseteq IC^s$, we have $L \subseteq [a] \cap IC^s$. The following claim establishes the required algebraic bounds for condition (4).

The following claim is the analog of Claim 7.12.

Claim 8.25. *The following hold:*

- $\sum_{\ell \in L} w(\ell) \leq \varepsilon \cdot \sum_{\ell \in [a+1:s]} w(\ell)$;
- *for any* $j \in DC^s$ *and* $\ell \in IC^F \setminus L$, *we have* $\alpha_\ell^c \leq (1 + 20\sqrt{\varepsilon}) \cdot (r_{js} + d(j, \ell))^2$;
- *for any* $j \in DC^s$ *and* $\ell \in [a+1:s]$, *we have* $\alpha_\ell^c \leq (r_{js} + d(j, \ell))^2$;
- *for any* $j \in DC^s$, $\ell \in L$, *and* $\ell' \in [a+1:s]$, *we have* $\alpha_\ell^c \leq (r_{js} + d(j, \ell'))^2$.

Proof. We prove the four bullets of the claim one by one.

First bullet. By the definition of Subcase 2(b), this weight bound is our direct assumption. Substituting the dummy variable ℓ directly yields the first bullet.

Second bullet. Consider any $j \in DC^s$ and $\ell \in IC^F \setminus L$. Let θ be the active dual parameter right before i_0 is opened in $\mathcal{H}'$. Because $\ell \notin L$, its distance to i satisfies $d(\ell, i) \leq \sqrt{\varepsilon\theta}$. Because i_0 freezes i, we know $d(i, i_0) \leq 2\sqrt{\varepsilon\theta}$. By the metric triangle inequality, $d(\ell, i_0) \leq d(\ell, i) + d(i, i_0) \leq 3\sqrt{\varepsilon\theta}$.

To bridge the distance to the open facilities S^s, we instantiate an active client $\ell' \in [a+1 : s] \cap B(i_0, \sqrt{\varepsilon\theta})$, which is guaranteed to exist by the Subcase 2(b) definition. Because ℓ' is active, we strictly know $\theta \leq (1-\delta) d^2(\ell', S^s)$, which geometrically implies $d(\ell', S^s) \geq \sqrt{\theta}$.

Applying the triangle inequality across ℓ, ℓ', and S^s yields the core distance relationship:

$$d(\ell, S^s) \geq d(\ell', S^s) - d(\ell, \ell') \geq d(\ell', S^s) - d(\ell, i_0) - d(i_0, \ell') \geq \sqrt{\theta} - 4\sqrt{\varepsilon\theta} \ .$$

Applying the triangle inequality to j, ℓ, and S^s, and scaling by $\sqrt{1-\delta}$, we directly obtain (because $\delta = \varepsilon/2$):

$$(r_{js} + d(j, \ell))^2 \geq (1-\delta) d^2(\ell, S^s) \geq \theta(1-\delta) \left(1 - 4\sqrt{\varepsilon}\right)^2 \geq \theta \left(1 - \varepsilon/2\right) \left(1 - 8\sqrt{\varepsilon}\right) > \theta(1 - 9\sqrt{\varepsilon}) \ .$$

Squaring both sides and applying the Taylor expansion $(1 - 9\sqrt{\varepsilon})^{-1} \leq 1 + 10\sqrt{\varepsilon}$ (valid for our sufficiently small $\varepsilon \in (0, 10^{-100})$) provides the exact upper bound for θ:

$$\theta \leq (1 + 10\sqrt{\varepsilon})(r_{js} + d(j, \ell))^2 \ .$$

Because $\ell \in IC^F$, it is indirectly connected right before i_0 opens, meaning its modified dual value is strictly bounded by the active parameter: $\alpha_\ell^c \leq \alpha_\ell \leq \theta$. Substituting our geometric bound yields the final inequality:

$$\alpha_\ell^c \leq (1 + 20\sqrt{\varepsilon})(r_{js} + d(j, \ell))^2 \ .$$

Third bullet. Consider any $j \in DC^s$ and $\ell \in [a+1:s] = A^F$. Because ℓ is active right before i_0 is opened, its maximum possible modified dual value is dictated by the second bullet of the η-openability condition (Definition 8.2), which strictly enforces $\tau_\ell \le (1-\delta)d^2(\ell, S^s)$. By Claim 8.20, we bypass the parameter θ entirely:

$$\alpha_\ell^c \le \tau_\ell \le (1-\delta)d^2(\ell, S^s) \ .$$

Applying the triangle inequality to j, ℓ, and S^s, we have $d(\ell, S^s) \le d(j, S^s) + d(j, \ell)$. Multiplying by $\sqrt{1-\delta}$ and substituting $r_{js} = \sqrt{1-\delta}d(j, S^s)$ yields:

$$\sqrt{1-\delta}d(\ell, S^s) \le r_{js} + \sqrt{1-\delta}d(j, \ell) \le r_{js} + d(j, \ell) \ .$$

Squaring both sides and substituting into our openability bound gives exactly:

$$\alpha_\ell^c \le (1-\delta)d^2(\ell, S^s) \le (r_{js} + d(j, \ell))^2 \ ,$$

completing the proof of the third bullet.

Fourth bullet. Consider any $j \in DC^s$, $\ell \in L$, and $\ell' \in [a+1:s]$. Because ℓ' is active right before i_0 is opened, we strictly have $\theta \le (1-\delta)d^2(\ell', S^s)$.

Applying the triangle inequality to j, ℓ', and S^s, we have $d(\ell', S^s) \le d(j, S^s) + d(j, \ell')$. Multiplying by $\sqrt{1-\delta}$ yields:

$$\sqrt{1-\delta}d(\ell', S^s) \le r_{js} + \sqrt{1-\delta}d(j, \ell') \le r_{js} + d(j, \ell') \ .$$

Squaring both sides provides the geometric upper bound for θ:

$$\theta \le (1-\delta)d^2(\ell', S^s) \le (r_{js} + d(j, \ell'))^2 \ .$$

Finally, because $\ell \in L \subseteq IC^F$, it is strictly indirectly connected before i_0 opens. According to the algorithm, its dual value is never increased during Stage 1 updates, ensuring $\alpha_\ell^c \le \alpha_\ell \le \theta$. Combining this with our geometric upper bound gives the exact final relation:

$$\alpha_\ell^c \le \theta \le (r_{js} + d(j, \ell'))^2 \ ,$$

completing the proof of the fourth bullet. □

Because Claim 8.25 perfectly satisfies condition (4) of Lemma 5.20 (since $IC^F \setminus L \cup [a+1:s] = IC^s \setminus L$, the second and third bullets map to the second condition of the lemma, while the fourth bullet maps directly to the third condition), we can apply Lemma 5.20.

Replacing the capacity bound $\hat{f}$ with $\hat{f} + 8n\eta$ throughout the linear combination evaluated in the lemma, we obtain:

$$\sum_{j \in [s]} w(j) \cdot (\alpha_j^c - (\rho-1)R_j^2) \le \hat{f} + 8n\eta + (\Gamma + 200\sqrt{\varepsilon}) \cdot \sum_{j \in [s]} w(j) \cdot d^2(j, i).$$

This identically satisfies our targeted dual feasibility bound required for the robust algorithm. This completes the proof of Lemma 8.16 for the Euclidean metric, and therefore the proof of Theorem 8.7.

9 $(4 + O(\sqrt{\varepsilon}))$-Approximation for $(\frac{\zeta}{\log n})$-Stable Instances

AS: I think many labels in Sections 9 and 10 are the same, and are creating some wrong references, specially by the end of this section.

AS: tried to rename the clashing labels in Section 10. check.

In this section, we give a $(4 + O(\sqrt{\varepsilon}))$-approximation algorithm for $(\frac{\zeta}{\log n})$-stable metric k-means instances, hence proving Theorem 1.4. We will reuse most of the notation and proofs in [CCG+25], and only mention the main differences.

To simplify notation, we let ε be the minimum of ε and ζ in the above theorem and further assume that $\varepsilon < 1/12$ is sufficiently small. With this notation, we present a $(4 + O(\sqrt{\varepsilon})$-approximation algorithm for $\varepsilon/\log(n)$-stable instances. This implies the above theorem as any $\zeta/\log(n)$-stable instance is also $\zeta'/\log(n)$-stable with $\zeta' \leq \zeta$.

The algorithm and the proof of the above result follow several steps where we iteratively simplify the instance by adding more structure. Finally, we obtain enough structure to solve the problem by maximizing a submodular function subject to a partition matroid constraint. We have not optimized constants in favor of simplifying the description, and throughout this section we let $O_\varepsilon(\log n)$ denote $O(\log(n)/\varepsilon^{O(1)})$.

Definitions of leaders. We use the following definitions throughout this section. Given a set of centers $A \subseteq F$, we let $\text{cost}(A) := \sum_{p \in D} d^2(p, A)$ denote the cost of the corresponding clustering (where each client p is assigned to the closest center in A and pays the corresponding distance). Sometimes we will consider a (possibly suboptimal) assignment $\mu : D \to A$ of clients to centers in A, and let $\text{cost}(A, \mu) := \sum_{p \in D} d^2(p, \mu(p))$ be the corresponding cost. Let OPT be a fixed optimum solution for the k-means instance, and let $\text{opt} = \text{cost}(\text{OPT})$ be its cost. For a client $p \in D$, let $\text{OPT}_p := d^2(p, \text{OPT})$ be the cost paid by p in OPT.

Let C^* be an OPT cluster with center $c^* \in \text{OPT}$. We let $\text{avg}_{C^*,\text{OPT}}$ be the squared distance from c^* to the $\varepsilon|C^*|$th closest client of C^* to c^*. We further let C^*_{avg} be the set of clients of C^* at a squared distance at most $\text{avg}_{C^*,\text{OPT}}$ from c^*, we refer to these clients as *leaders* for C^*. So $\text{avg}_{C^*,\text{OPT}}$ is the *maximum squared distance* from a leader to its center c^*. Also note that we have, $|C^*_{avg}| \geq \varepsilon|C^*|$. We thus have that $\text{avg}_{C^*,\text{OPT}}$ is upper bounded by $\frac{1}{1-\varepsilon}$ times the average cost $\frac{1}{|C^*|}\sum_{p \in C^*} d^2(p, c^*)$, and at the same time C^*_{avg} contains a constant fraction of the clients.

Partition matroids. In Subsection 9.10, we formulate a submodular function that we then maximize over a partition matroid using known results. We define those concepts and also state the result that we will use. First, recall the definition of a matroid and partition matroid. A *matroid* is a tuple $(E, \mathcal{I})$ defined on a ground set E with a family of independent sets $\mathcal{I}$ that satisfy: (1) if $A \in \mathcal{I}$ and $A' \subseteq A$, then $A' \in \mathcal{I}$; and (2) if $A, B \in \mathcal{I}$ and $|A| < |B|$, then there exists an element $e \in B - A$ such that $A \cup \{e\} \in \mathcal{I}$. In the special case of a *partition matroid*, the ground set E is partitioned into disjoint subsets $E_1, E_2, \ldots, E_k$, and there are non-negative integers $r_1, r_2, \ldots, r_k$ such that:

- A subset $I \subseteq E$ is independent if and only if $|I \cap E_i| \leq r_i$ for each $i = 1, 2, \ldots, k$.

In other words, each subset E_i has a capacity limit r_i, and an independent set I cannot contain more than r_i elements from E_i. To work with matroids (to get a running time that is polynomial in the size of the ground set E), we often use *independence queries*, which allow us to determine if a given subset $A \subseteq E$ is independent (i.e., whether $A \in \mathcal{I}$). Specifically, an independence query on a subset A returns true if A is independent and false otherwise.

Submodular functions. Having defined matroids, we proceed to define non-negative monotone submodular functions. For a finite ground set E, a set function $f : 2^E \to \mathbb{R}$ is *submodular* if it satisfies the diminishing returns property: for every pair of sets $A \subseteq B \subseteq E$ and any element $e \in E - B$, it holds that

$$f(A \cup \{e\}) - f(A) \geq f(B \cup \{e\}) - f(B).$$

Intuitively, this means that adding an element e to a smaller set A provides at least as much additional value as adding e to a larger set B. In addition, f is called *monotone* if for any pair of sets $A \subseteq B \subseteq E$, we have $f(A) \leq f(B)$. In other words, adding elements to a set does not decrease the function value. Finally, f is *non-negative* if $f(A) \geq 0$ for all $A \subseteq E$.

A celebrated result [CCPV11] gives a polynomial-time $(1 - 1/e)$-approximation algorithm for the problem of maximizing a non-negative monotone submodular function over a matroid constraint. While the original result was a randomized algorithm, a recent striking result [BF24], building upon the work of [FW14], gives a deterministic algorithm with the same guarantee. We remark that polynomial time here refers to a running time that is polynomial in the size of the ground set E. To summarize, they show the following theorem.

Theorem 9.1. *Let $f : 2^E \to \mathbb{R}$ be a non-negative monotone submodular function that we can evaluate in polynomial time, and let $(E, \mathcal{I})$ be a matroid on the same groundset, for which we can answer independence queries in polynomial time. Then, for every $\zeta > 0$, there is a polynomial-time algorithm that outputs a set $X \subseteq E$ with $X \in \mathcal{I}$ so that*

$$f(X) \geq (1 - 1/e - \zeta) \cdot \max_{X^* \in \mathcal{I}} f(X^*) .$$

9.1 Properties of a Locally Optimal Solution

Let S be a locally optimal solution output by the following standard local search algorithm. We remark that its running time is polynomial since by Lemma 3.2 the cost of a solution is a polynomially bounded integer and, at each improving step, the cost decreases by at least one.

Classic Local Search Procedure for k-Means.

1. $S \leftarrow$ Arbitrary solution for k-Means.
2. While there exists a solution S' such that $|S \Delta S'| \leq 2$ and $\text{cost}(S') < \text{cost}(S)$:
   (a) $S \leftarrow S'$.
3. Output S.

Local search achieves a 26 approximation for k-Means, see [GT08].

Lemma 9.2. *$cost(S) \leq 26 \cdot opt$.*

In the remaining part of the algorithm and its analysis we fix S, and recall that we fixed the reference optimal solution OPT. We distinguish different types of clusters of OPT and analyze their properties in the next two sections. We let $S_p := d^2(p, S)$ be the cost paid by p in S and recall that OPT_p denotes the cost $p \in D$ pays in OPT.

9.2 Pure Clusters

We say that a cluster C^* of OPT is *pure* (w.r.t. S) if there exists a cluster C' of S such that $|C' \Delta C^*| \leq 3\varepsilon \min(|C^*|, |C'|)$. In which case we also say that C' is pure w.r.t. OPT. We say that C' and C^* are associated. The following lemma follows from the fact that the input is β-stable for $\beta := \varepsilon/\log n$ and that the solution S is a 26-approximation by Lemma 9.2. Given a client or center a and a distance r, we let $B(a, r)$ be the set of clients and centers at distance at most r from a.

Lemma 9.3. *[CCG+25] The number k_{imp} of clusters of OPT that are not pure w.r.t. S (and thus the number of clusters of S which are not pure w.r.t OPT) is at most $\frac{\log n}{\varepsilon^3}$.*

Let C^* be a pure cluster of OPT and C' be the associated cluster of S. Let c^* be the center of C^* and c' be the center of C'. We define $t(c^*) := c'$. For any client $p \in C^*$, let $r(p) := d^2(p, c') = d^2(p, t(c^*))$. We have the following lemma which is the only lemma that requires that the solution S is obtained via the local search (the proof of the previous lemma only used that it was a constant-factor approximation). In words, it says that S approximates the connection cost of clients belonging to pure clusters of the optimal solution almost perfectly.

Lemma 9.4. *[CCG+25] Let $D^*_{pure} \subseteq D$ be the subset of clients that belong to pure clusters of the optimal solution OPT. We have*

$$\sum_{p \in D^*_{pure}} r(p) \leq \sum_{p \in D^*_{pure}} OPT_p + O(\sqrt{\varepsilon} \cdot opt)\,.$$

9.3 Cheap Clusters and D-Sample Process to Hit Expensive Clusters

In the previous section, we showed that the cost of pure clusters in OPT is close to their optimal cost in our local search solution S. Our goal in this section is to "hit" those non-pure clusters of OPT that have a high cost in S. We say that a non-pure cluster C^* of OPT with center c^* is *basic-cheap* if the total cost in S of the clients in C^*_{avg} is less than $\varepsilon^5 \text{opt}/\log n$ or if there exists a center of S at squared distance at most $(1+\varepsilon)\text{avg}_{C^*,\text{OPT}} + \varepsilon\gamma_{C^*}$ from c^*, where $\gamma_{C^*} := \frac{1}{|C^*|}\sum_{p \in C^*}(\text{OPT}_p + S_p)$; in the latter case we also say that C^* is *covered* by S.

The following procedure, that aims to "hit" all clusters of OPT that are non-pure and non-basic-cheap, is inspired by the classic k-means++ algorithm and its so called D^2-sampling procedure to sample clients proportional to their cost in solution S.

s-D^2-Sample Process.

1. Let W be the set obtained by taking s independent samples of clients where, in each sample, client p is sampled with probability $\frac{d^2(p,S)}{\text{cost}(S)}$.
2. Output W.

Our algorithm obtains a set W of clients by sampling s^* using the s^*-D^2-sample process, where

$$s^* = 26 \cdot \frac{\log n}{\varepsilon^5} \ln\left(\frac{81}{\varepsilon^2}\right)\,.$$

We have the following lemma that says that we hit all non-pure and non-basic-cheap clusters except for a set V of clusters that have an insignificant cost in S.

For a non-pure cluster C^* of OPT with center $c^* \in \text{OPT}$, we let $t(c^*)$ be the closest center in S to c^*, and for any $p \in C^*$, we let the replacement cost of p be $r(p) := d^2(p, t(c^*))$.

Lemma 9.5. *[CCG+25] Consider the random set W of clients obtained from the s^*-D^2-sample process. Let V be the set of clusters C^* of OPT that are non-pure and non-basic-cheap such that $C^*_{avg} \cap W = \emptyset$. With probability at least $1 - \varepsilon$,*

$$\sum_{C^* \in V} \sum_{p \in C^*} r(p) \leq \varepsilon \cdot opt. \tag{45}$$

Successful W. We say that the set of clients W obtained from the s^*-D^2-sample process is a *successful* sample if (45) holds. We define the following two types of OPT clusters. We say that a non-pure cluster C^* of OPT with center c^* is *cheap* if it is basic-cheap or is in V; a non-pure cluster is *expensive* otherwise. In other words, a cluster C^* of the optimal solution is expensive if it is a non-pure and non-basic-cheap cluster such that $C^*_{avg} \cap W \neq \emptyset$.

Assuming that W is successful, we have a good bound on the cost of the clients assigned to cheap centers in the optimal solution.

Lemma 9.6. *[CCG+25] Let $D^*_{\texttt{cheap}} \subseteq D$ be the subset of clients that belong to cheap clusters of the optimal solution OPT. If W is a successful sample,*

$$\sum_{p \in D^*_{cheap}} r(p) \leq \sum_{p \in D^*_{cheap}} 4\, OPT_p + O(\sqrt{\varepsilon} \cdot opt)\,.$$

Given S and W, the set of clusters of the optimal solution OPT is thus partitioned into pure clusters, cheap clusters, and expensive clusters. We let $\hat{\text{OPT}}$ be the centers of OPT that are expensive. We further define the set $D^*_{\texttt{exp}} \subseteq D$ to be the subset of clients that belong to the expensive clusters in OPT. We have thus partitioned D into sets $D^*_{\texttt{pure}}, D^*_{\texttt{cheap}}$, and $D^*_{\texttt{exp}}$ depending on the type of optimal cluster they belong to. Lemma 9.4 bounds the cost of the clients in $D^*_{\texttt{pure}}$ in S and Lemma 9.6 bounds the cost of the clients in $D^*_{\texttt{cheap}}$. For future reference, we summarize these two lemmas (by weakening the upper bound for clients in $D^*_{\texttt{pure}}$).

AS: i think there's duplication of labels across euclidean and this section

AS: fine now.

Lemma 9.7. *Assuming W is a successful sample,*

$$\sum_{p \in D - D^*_{exp}} r(p) \leq \sum_{p \in D - D^*_{exp}} 4 OPT_p + O(\sqrt{\varepsilon}) \cdot opt\,.$$

It follows that the connection cost in solution S of clients in $D - D^*_{\texttt{exp}}$ is roughly within a factor 4 of their cost in the optimal solution. The remaining part of the algorithm is thus devoted to modifying S to obtain a small connection cost of the clients in $D^*_{\texttt{exp}}$ without significantly increasing the cost of the other clients.

9.4 Identifying Balls of Expensive Clusters

By the definition of expensive clusters, each expensive cluster C^* of OPT satisfies $C^* \cap W \neq \emptyset$ where W is the sample obtained by running the sampling procedure of the last subsection. Here, our goal is to guess a subset $\mathcal{B}$ of balls so that for each expensive cluster C^* there is a ball $B(\ell, \sqrt{\rho}) \in \mathcal{B}$ so that ℓ, which we refer to as a leader, is in C^*_{avg} and ρ is a very close approximation to $\text{avg}_{C^*,\text{OPT}}$. This guarantees that the center $c^* \in \hat{\text{OPT}}$ of cluster C^* is in $B(\ell, \sqrt{\rho})$ and that no other center in $B(\ell, \sqrt{\rho})$ is "too" far from c^* since the radius of the ball is $\approx \sqrt{\text{avg}_{C^*,\text{OPT}}}$. The balls $\mathcal{B}$ will then be used when we approximate the centers in $\hat{\text{OPT}}$

via submodular function maximization subject to a partition matroid constraint (the balls will correspond to the partitions of the matroid). As the algorithm does not know the set of expensive clusters, it enumerates all subsets of W, and for each guessed point, it enumerates a constant number of radii.

We say that a set of balls $\mathcal{B}$ is a *valid* set of balls if it satisfies the following: For each expensive cluster C^* of OPT, we have a ball $B(\ell, \sqrt{\rho}) \in \mathcal{B}$ such that $\ell \in C^*_{avg}$ is a leader and

$$\mathrm{avg}_{C^*,\mathrm{OPT}} \leq \rho \leq \mathrm{avg}_{C^*,\mathrm{OPT}} + \varepsilon\gamma_{C^*},$$

where we recall that $\gamma_{C^*} = \frac{1}{|C^*|}\sum_{p\in C^*}(\mathrm{OPT}_p + S_p)$.

Lemma 9.8. *[CCG+25] In polynomial time $n^{\varepsilon^{-O(1)}}$ one can compute a collection $\mathcal{L}_{\mathtt{bal}}$ of sets of balls such that at least one set $\mathcal{B} \in \mathcal{L}_{\mathtt{bal}}$ is a valid set of balls.*

While our algorithm will try all possible sets in $\mathcal{L}_{\mathtt{bal}}$ (since it does not know which one is valid), we do our analysis by considering the run of the algorithm when it selects this set $\mathcal{B}$ of valid balls.

9.5 *Dummy* Centers and Mixed Solutions M_O and M_D

Next, the algorithm creates, for each ball $B(\ell, \sqrt{\rho}) \in \mathcal{B}$ with center ℓ and radius $\sqrt{\rho}$, a *dummy* center δ. The center δ is at distance $\sqrt{\rho}$ from ℓ and at distance $\sqrt{\rho} + d(\ell, p)$ from any other input point p.

Let Λ be the set of dummy centers, so $|\Lambda| = |\mathcal{B}|$. Recall that $D^*_{\mathtt{exp}}$ is the set of clients that are in an expensive cluster of OPT, i.e., served by a center in $\hat{\mathrm{OPT}}$ in the optimal solution.

Lemma 9.9. *[CCG+25] Suppose the sample W is successful, and that the set of balls $\mathcal{B}$ is valid, then there exists a set of centers S_0 of S that satisfies the following properties:*

1. *$|S_0| = |\hat{OPT}| \leq \frac{\log n}{\varepsilon^3}$, and;*
2. *$\forall c \in S_0$, there is no $c^* \in OPT - \hat{OPT}$ such that $t(c^*) = c$, and;*
3. *$cost(S - S_0 \cup \Lambda) \leq 9\sum_{p\in D^*_{exp}} OPT_p + \sum_{p\in D - D^*_{exp}} r(p) + O(\varepsilon \cdot opt)) \leq (9 + O(\sqrt{\varepsilon}))opt.$*

Definition of the mixed solutions M_O and M_D. Next, let us analyze the cost of the swap $(\hat{\mathrm{OPT}}, S_0)$. Namely, of the solution $M_O = S - S_0 \cup \hat{\mathrm{OPT}}$, where M in M_O stands for "mixed solution" and the subscript O stands for that the mixed solution is obtained by swapping in some elements from OPT (and removing S_0). We will also analyze the "dummy" version of M_O where instead of swapping in $\hat{\mathrm{OPT}}$, we swap in the dummy centers Λ. We denote that solution by $M_D = S - S_0 \cup \Lambda$, where subscript D stands for "dummy".

Lemma 9.10. *[CCG+25] We have $d^2(p, \hat{OPT}) \leq OPT_p$ if $p \in D^*_{exp}$ and $d^2(p, S - S_0) \leq r(p)$ if $p \in D - D^*_{exp}$.*

Notice that M_O contains $S - S_0$ and $\hat{\mathrm{OPT}}$. Therefore, if we sum up the above bounds for all clients we get

$$\mathrm{cost}(M_O) \leq \sum_{p\in D^*_{\mathtt{exp}}} \mathrm{OPT}_p + \sum_{p\in D - D^*_{\mathtt{exp}}} r(p)\,,$$

which by Lemma 9.7 is at most $\sum_{p\in D^*_{\mathsf{exp}}} \mathrm{OPT}_p + \sum_{p\in D-D^*_{\mathsf{exp}}} 4\mathrm{OPT}_p + O(\sqrt{\varepsilon})\cdot\mathrm{opt}$ (assuming that that the sample W was successful). By swapping S_0 with $\hat{\mathrm{OPT}}$ we thus get a 1-approximation on the clients D^*_{exp} and a 4-approximation of the remaining clients (plus a small error term). Our goal in the next sections will thus be to approximate this swap. The next few steps will be aimed at simplifying (or guessing parts of) S_0. The last step will then approximate $\hat{\mathrm{OPT}}$ and the remaining part of S_0 by submodular function maximization subject to a partition matroid constraint. For that part, it will be helpful to have a bound on the dummy solution as well; specifically, a modified version of it (see Lemma 9.12 and Claim 9.15). For intuition of those statements, let us here say that one can show

$$\mathrm{cost}(M_D) \le \sum_{p\in D^*_{\mathsf{exp}}} 9\,\mathrm{OPT}_p + \sum_{p\in D-D^*_{\mathsf{exp}}} r(p) + O(\sqrt{\varepsilon})\cdot\mathrm{opt}$$

by observing that only the clients D^*_{exp} have a different connection cost in M_D than in M_O. Moreover, each such client $p \in D^*_{\mathsf{exp}}$ that was previously assigned to a center $c^* \in \hat{\mathrm{OPT}}$, and belongs to cluster C^* of OPT with leader ℓ, has an associated dummy center δ at squared distance

$$d^2(p,\delta) \le 3d^2(p,c^*) + 3d^2(c^*,\ell) + 3d^2(\ell,\delta) \le 3d^2(p,c^*) + 6\mathrm{avg}_{C^*,\mathrm{OPT}} + 3\varepsilon\gamma_{C^*}$$

(assuming the set $\mathcal{B}$ of balls is valid and using Lemma 3.5 with $\gamma = 3$). Simplifications then give the stated inequality as $|C^*|\mathrm{avg}_{C^*,\mathrm{OPT}} \le (1+2\varepsilon)\sum_{p\in C^*}\mathrm{OPT}_p$ and $|C^*|\varepsilon\gamma_{C^*} = \varepsilon\sum_{p\in C^*}(\mathrm{OPT}_p + S_p)$.

9.6 Removal of Expensive Centers of S_0

Thus, it remains to show that our algorithm can yield a good approximation to the gain that the swap $(\hat{\mathrm{OPT}}, S_0)$ would provide. Of course, it will not necessarily be optimum, but we can show it is enough to conclude the proof of our theorem. We focus on the iteration of the algorithm that considers a valid ball set of balls $\mathcal{B}$. In particular, at this point given that W is successful and $\mathcal{B}$ valid, the number of balls in our ball guess is equal to $|S_0| = |\hat{\mathrm{OPT}}| = |\Lambda| \le \frac{\log n}{\varepsilon^3}$. Our algorithm then makes use of the following Lemma in [CCG+25] that takes as input the local search solution S and the set Λ of dummy centers.

Lemma 9.11. *[CCG+25] There is a procedure that runs in $n^{1/\varepsilon^{O(1)}}$-time and produces a collection $\mathcal{L}_{exp}$ of at most $n^{1/\varepsilon^{O(1)}}$ subsets of S such that with probability at least $1-1/n$ there exists $\mathcal{Q}\in\mathcal{L}_{exp}$ satisfying*

1. *$\mathcal{Q}\subseteq S_0$, and*
2. *The total cost in the solution $S - Q\cup\Lambda$ of the clusters with centers in $S_0 - \mathcal{Q}$ is at most $\varepsilon\cdot opt$.*

9.7 Consistently Assigning Clients in Mixed Solutions and the sets $\mathcal{U}$ and $\mathcal{R}$

While our algorithm tries all possible $\mathcal{Q}\in\mathcal{L}_{\mathrm{exp}}$, we focus on the execution path of a set $\mathcal{Q}$ of centers satisfying the properties of Lemma 9.11, i.e.,

1. $\mathcal{Q}\subseteq S_0$, and

2. the cost in the solution $S - \mathcal{Q} \cup \Lambda$ of the clusters with centers in $S_0 - \mathcal{Q}$ is at most $\varepsilon \cdot \mathrm{opt}$.

We define the solution $S_{\mathcal{Q}} := S - \mathcal{Q} \cup \Lambda$ and for a center $c \in S_{\mathcal{Q}}$ we let $S_{\mathcal{Q}}(c)$ be the subset of clients closest to c (i.e., assigned to c) in the solution $S_{\mathcal{Q}}$. The second property above allows us to consider each cluster $S_{\mathcal{Q}}(c)$, $c \in S_0 - \mathcal{Q}$, as contracted by paying a small extra cost of $\varepsilon \cdot \mathrm{opt}$. In particular, this allows us to give a "consistent" assignment μ_O of clients in the mixed solutions. Here, consistent means that all clients in $S_{\mathcal{Q}}(c)$, for $c \in S_0 - \mathcal{Q}$, are assigned to the same center in the mixed solutions where all of S_0 is removed.

Definition of μ_O and μ_D. In the following we slightly deviate from the approach in [CCG+25]. More specifically, the constant 4 below is a 5 in [CCG+25]. The rest of the analysis remains however very similar. We modify how he clients are assigned in the solution M_O to obtain the assignment μ_O. For every $c \in S_0 - \mathcal{Q}$, all the clients in $S_{\mathcal{Q}}(c)$ are reassigned to the same center of M_O as follows. Let c_1 be the cluster of $\hat{\mathrm{OPT}}$ that is the closest to c and let c_2 be the center of $M_O - \hat{\mathrm{OPT}} = S - S_0$ that is the closest to c. The clients of $S_{\mathcal{Q}}(c)$ are all assigned to either c_1 or c_2:

F: Changed the constant from 5 to 4

- If $d^2(c, c_1) \leq d^2(c, c_2)/4$, assign $S_{\mathcal{Q}}(c)$ to c_1; $S_{\mathcal{Q}}(c)$ is called a *type-1* cluster.
- Otherwise ($d^2(c, c_1) > d^2(c, c_2)/4$), assign $S_{\mathcal{Q}}(c)$ to c_2; $S_{\mathcal{Q}}(c)$ is called a *type-2* cluster.

Let T_1 be the set of clients in a type-1 cluster and T_2 the set of clients in a type-2 cluster. The remaining clients in $D^*_{\mathsf{exp}} - (T_1 \cup T_2)$ are assigned to their closest centers in $\hat{\mathrm{OPT}}$, and the remaining clients in $(D - D^*_{\mathsf{exp}}) - (T_1 \cup T_2)$ are assigned to their closest centers in $S - S_0$. Notice that, even neglecting the extra cost due to the consistent assignment, the above assignment of clients in T_2 is suboptimal when $d(c, c_2) > d(c, c_1) > d(c, c_2)/4$. The motivation for this reassignment is technical. More specifically, it will be used in Lemma 9.12 to have a better upper bound on the cost of clients in $C^* \cap T_1$ for a non-expensive cluster C^* (case b2). This leads to a larger upper bound on the cost of clients in $C^* \cap T_2$ for an expensive cluster C^* (case a3), which is however tolerable.

We further obtain an assignment μ_D of clients in M_D by modifying μ_O as follows: each client p with $\mu_O(p) \in \hat{\mathrm{OPT}}$ is assigned to the associated dummy center by μ_D. So for each client p we have $\mu_D(p) = \mu_O(p)$ unless $\mu_O(p) \in \hat{\mathrm{OPT}}$ in which case $\mu_D(p)$ equals the dummy center associated with $\mu_O(p)$.

Sets $\mathcal{R}$ and $\mathcal{U}$. By the definition of type-1 and type-2 cluster, we can split the set of centers of $S_0 - \mathcal{Q}$ into two groups $\mathcal{R}$ and $\mathcal{U}$. Let $\mathcal{R}$ be the set of centers of $S_0 - \mathcal{Q}$ whose set of clients is completely assigned to a center of $\hat{\mathrm{OPT}}$ in the assignment μ_O, and let $\mathcal{U} = S_0 - \mathcal{Q} - \mathcal{R}$ be the remaining ones that are assigned to centers in $S - S_0$.

We end this section by analyzing the costs of the assignments μ_O and μ_D. Recall that we use the notation $\mathrm{cost}(M_O, \mu_O)$ to denote the cost of M_O equipped with assignment μ_O and, similarly $\mathrm{cost}(M_D, \mu_D)$ for the cost of M_D equipped with assignment μ_D. We say that the selection of $(W, \mathcal{B}, \mathcal{Q})$ is successful, if the sample W selected in Subsection 9.3 is successful, the set of balls $\mathcal{B}$ from Subsection 9.4 is valid, and $\mathcal{Q}$ selected in this section satisfies the properties of Lemma 9.11. In [CCG+25] there is an analogue version of the next lemma, where (besides the mentioned difference in the constant 4 versus 5) only the sum $\mathrm{cost}(M_O, \mu_O) + \mathrm{cost}(M_D, \mu_D)$ is upper bounded. For reasons that will become clearer later, we need to bound the two quantities separately.

Lemma 9.12. *If the selection of $(W, \mathcal{B}, \mathcal{Q})$ is successful, one has*

F: Bounded separately the two terms, as needed later

$$\begin{aligned} cost(M_O, \mu_O) \leq & \sum_{p \in D^*_{exp} - T_2} OPT_p + \sum_{p \in D^*_{exp} \cap T_2} 4\mathrm{OPT}_p \\ & + \sum_{p \in (D - D^*_{exp}) - T_1} r(p) + \sum_{p \in (D - D^*_{exp}) \cap T_1} \frac{1}{4} r(p) + O(\sqrt{\varepsilon}) opt \leq O(1) opt, \end{aligned}$$

and

$$\begin{aligned} cost(M_D, \mu_D) \leq & \sum_{p \in D^*_{exp} - T_2} 9OPT_p + \sum_{p \in D^*_{exp} \cap T_2} 4\mathrm{OPT}_p \\ & + \sum_{p \in (D - D^*_{exp}) - T_1} r(p) + \sum_{p \in (D - D^*_{exp}) \cap T_1} \frac{9}{4} r(p) + O(\sqrt{\varepsilon}) opt \leq O(1) opt. \end{aligned}$$

Proof. Throughout the proof of the lemma, we will repeatedly upper bound $d^2(p, \mu_D(p))$ in terms of $d^2(p, \mu_O(p))$ using the following bounds:

Claim 9.13. *[CCG+25] Consider a client $p \in S_{\mathcal{Q}}(c)$ with $\mu_O(p) \neq \mu_D(p)$ and so $\mu_O(p) \in \hat{OPT}$ is the center of an expensive cluster C^* of OPT. Then*

1. $d^2(p, \mu_D(p)) \leq 3d^2(p, \mu_O(p)) + 6(avg_{C^*, OPT} + \varepsilon \gamma_{C^*})$, *and*
2. $d^2(p, \mu_D(p)) \leq \frac{2}{\sqrt{\varepsilon}} d^2(p, S_{\mathcal{Q}}) + 9(1 + \sqrt{\varepsilon}) d^2(c, \mu_O(p))$.

We prove the lemma by distinguishing between expensive and non-expensive clusters C^* of OPT.

Case a: C^* is an expensive cluster of OPT. Let $c^* \in \hat{\mathrm{OPT}}$ be the center of C^*. Let $p \in C^*$ be a client in C^* and let c be its closest center in $S_{\mathcal{Q}}$, i.e., $p \in S_{\mathcal{Q}}(c)$. We bound $d^2(p, \mu_O(p))$ and $d^2(p, \mu_D(p))$ by considering three cases:

Case a1: $p \in C^* - (T_1 \cup T_2)$. By definition, we have $\mu_O(p)$ is the closest center in $\hat{\mathrm{OPT}}$. Moreover, $\mu_O(p)$ equals the cluster center c^* of C^* since p is closest to c^* among all centers in $\mathrm{OPT} \supseteq \hat{\mathrm{OPT}}$. Lemma 9.10 gives us $d^2(p, \mu_O(p)) = \mathrm{OPT}_p$. Hence claim 9.13.1 implies,

$$d^2(p, \mu_D(p)) \leq 3\mathrm{OPT}_p + 6(\mathrm{avg}_{C^*, \mathrm{OPT}} + \varepsilon \gamma_{C^*}).$$

Case a2: $p \in C^* \cap T_1$. By definition of type-1 clusters, $\mu_O(p) = c_1$ is the closest center in $\hat{\mathrm{OPT}}$ to c. Using Lemma 3.4 with $\gamma = 1 + \sqrt{\varepsilon}$ twice,

F: Just split previous sum in two parts

$$\begin{aligned} d^2(p, \mu_O(p)) &\leq \frac{2}{\sqrt{\varepsilon}} d^2(p, S_{\mathcal{Q}}) + (1 + \sqrt{\varepsilon}) d^2(c, c_1) \\ &\leq \frac{2}{\sqrt{\varepsilon}} d^2(p, S_{\mathcal{Q}}) + (1 + \sqrt{\varepsilon}) \left(\frac{2}{\sqrt{\varepsilon}} d^2(p, S_{\mathcal{Q}}) + (1 + \sqrt{\varepsilon}) d^2(p, c_1) \right) \\ &\leq \frac{5}{\sqrt{\varepsilon}} d^2(p, S_{\mathcal{Q}}) + (1 + O(\sqrt{\varepsilon}))\mathrm{OPT}_p. \end{aligned}$$

Furthermore, by Claim 9.13.2 and using similarly as above Lemma 3.4 with $\gamma = 1+\sqrt{\varepsilon}$,

$$\begin{aligned} d^2(p,\mu_D(p)) &\le \frac{2}{\sqrt{\varepsilon}} d^2(p,S_{\mathcal{Q}}) + 9(1+\sqrt{\varepsilon}) d^2(c,c_1) \\ &\le \frac{2}{\sqrt{\varepsilon}} d^2(p,S_{\mathcal{Q}}) + 9(1+\sqrt{\varepsilon})(\frac{2}{\sqrt{\varepsilon}} d^2(p,S_{\mathcal{Q}}) + (1+\sqrt{\varepsilon}) d^2(p,c_1)) \\ &\le \frac{21}{\sqrt{\varepsilon}} d^2(p,S_{\mathcal{Q}}) + (9+O(\sqrt{\varepsilon}))\mathrm{OPT}_p\,. \end{aligned}$$

Case a3: $p \in C^* \cap T_2$**.** By the definition of type-2 clusters, $\mu_O(p) = c_2$ is the closest center to c in $S - S_0$, and $d^2(c,c_2) \le 4d^2(c,\hat{\mathrm{OPT}})$. So, by Lemma 3.4 with $\gamma = 1+\sqrt{\varepsilon}$, $d^2(c,c_2) \le 4(\frac{2}{\sqrt{\varepsilon}} d^2(c,p) + (1+\sqrt{\varepsilon})\mathrm{OPT}_p)$ where we additionally used that $d^2(p,\hat{\mathrm{OPT}}) \le \mathrm{OPT}_p$ by Lemma 9.10. As $\mu_O(p) = \mu_D(p)$ in this case, and using again Lemma 3.4 with $\gamma = 1+\sqrt{\varepsilon}$,

F: Do we mean that the lemma gives = here?

$$\begin{aligned} d^2(p,\mu_O(p)) = d^2(p,\mu_D(p)) &\le \frac{2}{\sqrt{\varepsilon}} d^2(p,c) + (1+\sqrt{\varepsilon}) d^2(c,c_2) \\ &\le \frac{2}{\sqrt{\varepsilon}} d^2(p,c) + (1+\sqrt{\varepsilon}) 4(\frac{2}{\sqrt{\varepsilon}} d^2(c,p) + (1+\sqrt{\varepsilon})\mathrm{OPT}_p) \\ &\le \frac{11}{\sqrt{\varepsilon}} d^2(p,S_{\mathcal{Q}}) + (4+O(\sqrt{\varepsilon}))\mathrm{OPT}_p\,. \end{aligned}$$

By the above bounds, we have that

$$\begin{aligned} \sum_{p\in C^*} d^2(p,\mu_O(p)) \le & \sum_{p\in C^*-(T_1\cup T_2)} \mathrm{OPT}_p + O(\sqrt{\varepsilon}) \sum_{p\in C^*} \mathrm{OPT}_p \\ & + \sum_{p\in C^*\cap T_1} \left(\mathrm{OPT}_p + \frac{5}{\sqrt{\varepsilon}} d^2(p,S_{\mathcal{Q}})\right) + \sum_{p\in C^*\cap T_2} \left(4\mathrm{OPT}_p + \frac{11}{\sqrt{\varepsilon}} d^2(p,S_{\mathcal{Q}})\right). \end{aligned}$$

Furthermore

$$\begin{aligned} \sum_{p\in C^*} d^2(p,\mu_D(p)) \le & \sum_{p\in C^*-(T_1\cup T_2)} \left(3\mathrm{OPT}_p + 6(\mathrm{avg}_{C^*,\mathrm{OPT}} + \varepsilon\gamma_{C^*})\right) + O(\sqrt{\varepsilon}) \sum_{p\in C^*} \mathrm{OPT}_p \\ & + \sum_{p\in C^*\cap T_1} \left(9\mathrm{OPT}_p + \frac{21}{\sqrt{\varepsilon}} d^2(p,S_{\mathcal{Q}})\right) + \sum_{p\in C^*\cap T_2} \left(4\mathrm{OPT}_p + \frac{11}{\sqrt{\varepsilon}} d^2(p,S_{\mathcal{Q}})\right). \end{aligned}$$

To complete the analysis for an expensive cluster C^*, it remains to upper bound the terms $6\varepsilon\gamma_{C^*}$ and $6\mathrm{avg}_{C^*,\mathrm{OPT}}$. By definition we have $|C^* - (T_1 \cup T_2)| \cdot 6\varepsilon\gamma_{C^*} \le 6\varepsilon \sum_{p\in C^*}(\mathrm{OPT}_p + S_p)$. For the other term, we claim that

$$6|C^* - (T_1 \cup T_2)| \cdot \mathrm{avg}_{C^*,\mathrm{OPT}} \le \sum_{p\in C^*-(T_1\cup T_2)} 6\mathrm{OPT}_p + 7\varepsilon \cdot \sum_{p\in C^*} \mathrm{OPT}_p\,.$$

This holds because there are at least $(1-\varepsilon)|C^*|$ clients in C^* such that $\mathrm{OPT}_p \ge \mathrm{avg}_{C^*,\mathrm{OPT}}$, which implies

$$\begin{aligned} \sum_{p\in C^*-(T_1\cup T_2)} 6\mathrm{OPT}_p + 7\varepsilon \cdot \sum_{p\in C^*} \mathrm{OPT}_p &\ge 6(|C^* - (T_1\cup T_2)| - \varepsilon|C^*|) \cdot \mathrm{avg}_{C^*,\mathrm{OPT}} + 7\varepsilon(1-\varepsilon)|C^*| \cdot \mathrm{avg}_{C^*,\mathrm{OPT}} \\ &\ge 6|C^* - (T_1 \cup T_2)| \cdot \mathrm{avg}_{C^*,\mathrm{OPT}}\,. \end{aligned}$$

Combining this with a previous inequality, we obtain

$$\sum_{p\in C^*} d^2(p,\mu_D(p)) \le \sum_{p\in C^*-(T_1\cup T_2)} 9\mathrm{OPT}_p + O(\sqrt{\varepsilon})\sum_{p\in C^*}(\mathrm{OPT}_p + S_p)$$
$$+ \sum_{p\in C^*\cap T_1}\left(9\mathrm{OPT}_p + \frac{21}{\sqrt{\varepsilon}}d^2(p,S_{\mathcal{Q}})\right) + \sum_{p\in C^*\cap T_2}\left(4\mathrm{OPT}_p + \frac{11}{\sqrt{\varepsilon}}d^2(p,S_{\mathcal{Q}})\right).$$

.

Case b: C^* is a non-expensive cluster of OPT. We proceed with a similar analysis. Indeed, consider a client $p \in C^*$ and let c be its closest center in $S_{\mathcal{Q}}$, i.e., $p \in S_{\mathcal{Q}}(c)$. We again upper bound $d^2(p,\mu_O(p))$ and $d^2(p,\mu_D(p))$ by considering three cases:

Case b1: $p \in C^* - (T_1 \cup T_2)$. By the definition of μ_O, we have that $\mu_O(p)$ is the closest center in $S - S_0$ to p. So $\mu_D(p) = \mu_O(p)$ and one has by Lemma 9.10

$$d^2(p,\mu_O(p)) = d^2(p,\mu_D(p)) \le r(p).$$

Case b2: $p \in C^* \cap T_1$. By the definition of type-1 clusters, $\mu_O(p) = c_1$ is the closest center in $\hat{\mathrm{OPT}}$ to c. By Lemma 3.4 with $\gamma = 1+\sqrt{\varepsilon}$,

$$d^2(p,\mu_O(p)) \le \frac{2}{\sqrt{\varepsilon}}d^2(p,S_{\mathcal{Q}}) + (1+\sqrt{\varepsilon})d^2(c,c_1).$$

Furthermore, by Claim 9.13.2,

$$d^2(p,\mu_D(p)) \le \frac{2}{\sqrt{\varepsilon}}d^2(p,S_{\mathcal{Q}}) + 9(1+\sqrt{\varepsilon})d^2(c,c_1).$$

We proceed by upper bounding $d^2(c,c_1)$. Here we critically use the fact that, by the definition of T_1, one has $d^2(c,c_1) \le d^2(c,c_2)/4$ where c_2 is the closest center to c in $S - S_0$. Let c_3 be the closest center to p in $S - S_0$. Then, using Lemma 3.4 with $\gamma = 1+\sqrt{\varepsilon}$, we obtain that

$$d^2(c,c_1) \le \frac{1}{4}d^2(c,c_2) \le \frac{1}{4}d^2(c,c_3) \le \frac{2}{4\sqrt{\varepsilon}}d^2(p,S_{\mathcal{Q}}) + \frac{1+\sqrt{\varepsilon}}{4}d^2(p,c_3).$$

Moreover, we have $d^2(p,c_3) = d^2(p,S-S_0) \le r(p)$ by Lemma 9.10 and so $d^2(c,c_1) \le \frac{2}{4\sqrt{\varepsilon}}d^2(p,S_{\mathcal{Q}}) + \frac{1+\sqrt{\varepsilon}}{4}r(p)$, which gives us the bounds

$$d^2(p,\mu_O(p)) \le \frac{5}{\sqrt{\varepsilon}}d^2(p,S_{\mathcal{Q}}) + \frac{1+O(\sqrt{\varepsilon})}{4}r(p).$$

and

$$d^2(p,\mu_D(p)) \le \frac{21}{\sqrt{\varepsilon}}d^2(p,S_{\mathcal{Q}}) + \frac{9+O(\sqrt{\varepsilon})}{4}r(p).$$

Case b3: $p \in C^* \cap T_2$. By the definition of type-2 clusters, $\mu_O(p) = c_2$ is the closest center to c in $S - S_0$, and so $\mu_O(p) = \mu_D(p)$. Now by Lemma 9.10 and applying twice Lemma 3.4

with $\gamma = 1 + \sqrt{\varepsilon}$,

$$\begin{aligned} d^2(p,\mu_O(p)) = d^2(p,\mu_D(p)) &\le \frac{2}{\sqrt{\varepsilon}} d^2(p,c) + (1+\sqrt{\varepsilon}) d^2(c,c_2) \\ &\le \frac{2}{\sqrt{\varepsilon}} d^2(p,c) + (1+\sqrt{\varepsilon}) \left(\frac{2}{\sqrt{\varepsilon}} d^2(p,c) + (1+\sqrt{\varepsilon}) d^2(p,c_2) \right) \\ &\le \frac{5}{\sqrt{\varepsilon}} d^2(p,c) + (1+\sqrt{\varepsilon})^2 d^2(p, S - S_0) \\ &\le \frac{5}{\sqrt{\varepsilon}} d^2(p,S_{\mathcal{Q}}) + (1+O(\sqrt{\varepsilon})) r(p). \end{aligned}$$

Summing up the above bounds, for a non-expensive cluster C^* of OPT we get

$$\begin{aligned} \sum_{p \in C^*} d^2(p,\mu_O(p)) \le &\sum_{p \in C^* - (T_1 \cup T_2)} r(p) + \sum_{p \in C^* \cap T_1} \left(\frac{5}{\sqrt{\varepsilon}} d^2(p,S_{\mathcal{Q}}) + \frac{1+O(\sqrt{\varepsilon})}{4} r(p) \right) \\ &+ \sum_{p \in C^* \cap T_2} \left(\frac{5}{\sqrt{\varepsilon}} d^2(p,S_{\mathcal{Q}}) + (1+O(\sqrt{\varepsilon})) r(p) \right) \end{aligned}$$

and

$$\begin{aligned} \sum_{p \in C^*} d^2(p,\mu_D(p)) \le &\sum_{p \in C^* - (T_1 \cup T_2)} r(p) + \sum_{p \in C^* \cap T_1} \left(\frac{21}{\sqrt{\varepsilon}} d^2(p,S_{\mathcal{Q}}) + \frac{9+O(\sqrt{\varepsilon})}{4} r(p) \right) \\ &+ \sum_{p \in C^* \cap T_2} \left(\frac{5}{\sqrt{\varepsilon}} d^2(p,S_{\mathcal{Q}}) + (1+O(\sqrt{\varepsilon})) r(p) \right) \end{aligned}$$

If we sum up the above inequality for all non-expensive clusters and the bound obtained before for the expensive clusters of OPT we thus get

$$\begin{aligned} \mathrm{cost}(M_O,\mu_O) \le &\sum_{p \in D^*_{\mathrm{exp}} - T_2} \mathrm{OPT}_p + \sum_{p \in D^*_{\mathrm{exp}} \cap T_2} 4\mathrm{OPT}_p \\ &+ \sum_{p \in (D - D^*_{\mathrm{exp}}) - T_1} r(p) + \sum_{p \in (D - D^*_{\mathrm{exp}}) \cap T_1} \frac{1}{4} r(p) + O(\sqrt{\varepsilon})\mathrm{opt}\,, \end{aligned}$$

and

$$\begin{aligned} \mathrm{cost}(M_D,\mu_D) \le &\sum_{p \in D^*_{\mathrm{exp}} - T_2} 9\mathrm{OPT}_p + \sum_{p \in D^*_{\mathrm{exp}} \cap T_2} 4\mathrm{OPT}_p \\ &+ \sum_{p \in (D - D^*_{\mathrm{exp}}) - T_1} r(p) + \sum_{p \in (D - D^*_{\mathrm{exp}}) \cap T_1} \frac{9}{4} r(p) + O(\sqrt{\varepsilon})\mathrm{opt}\,. \end{aligned}$$

Above we used that $\sum_{c \in S_0 - \mathcal{Q}} \sum_{p \in S_{\mathcal{Q}}(c)} d^2(p,S_{\mathcal{Q}}) = \sum_{p \in T_1 \cup T_2} d^2(p,S_{\mathcal{Q}}) \le \varepsilon \cdot \mathrm{opt}$ (Lemma 9.11) and $\mathrm{cost}(S) \le 26\mathrm{opt}$ (Lemma 9.2) to bound the error term $O(\sqrt{\varepsilon} \cdot \mathrm{opt})$.

The global upper bound of $O(1)\mathrm{opt}$ simply follows in both cases from Lemma 9.7, which says that $\sum_{p \in D - D^*_{\mathrm{exp}}} r(p) \le \sum_{p \in D - D^*_{\mathrm{exp}}} 4\mathrm{OPT}_p + O(\sqrt{\varepsilon} \cdot \mathrm{opt})$ if the sample W is successful.

□

9.8 Removal of Cheap Centers in S_0

At this stage we have defined a clustering $S_{\mathcal{Q}} = S - \mathcal{Q} \cup \Lambda$, where Λ is the set of dummy centers. Recall that the mixed solution M_O equals $S - S_0 \cup \hat{\text{OPT}}$. So $S_{\mathcal{Q}} - \Lambda$ represents progress compared to S in that we have already removed a subset $\mathcal{Q}$ of S_0. Furthermore, with respect to the assignment of clients μ_O, we have that the remaining centers $S_0 - Q$, which we want to remove, are divided into two sets $\mathcal{R}$ and $\mathcal{U}$. The clusters with centers in $\mathcal{R}$ are completely assigned (by μ_O) to centers in $\hat{\text{OPT}}$, and every cluster with center in $\mathcal{U}$ is completely assigned to a center in $S - S_0$. The task of this section is to make further progress by guessing the centers in $\mathcal{U}$ and their reassignment to centers in $S - S_0$. This turns out to be a difficult task, and we will instead "approximately" guess a set $\widetilde{\mathcal{U}}$ and reassignment $\widetilde{\mu}$ of the clients of those clusters. Specifically, we will output a solution $S_{\mathcal{Q}\cup\widetilde{\mathcal{U}}} = S_{\mathcal{Q}} - \widetilde{\mathcal{U}}$ (obtained by removing $\widetilde{\mathcal{U}}$ from $S_{\mathcal{Q}}$) and an assignment of clients $\widetilde{\mu}$ to centers in $S_{\mathcal{Q}} - \widetilde{\mathcal{U}}$. This is done through the following lemma. Recall that for $c \in S_{\mathcal{Q}}$, $S_{\mathcal{Q}}(c)$ is the set of clients assigned to c in $S_{\mathcal{Q}}$, i.e., closest to c among all centers in $S_{\mathcal{Q}}$.

Lemma 9.14. *[CCG+25] Given S and $\mathcal{Q}$ as described above, there is a polynomial-time algorithm that produces a collection $\mathcal{L}_{cheap}$ of subsets $\widetilde{\mathcal{U}}$ of $S - \mathcal{Q}$, and for each such $\widetilde{\mathcal{U}}$ and assignment $\widetilde{\mu}$ of clients to centers in $S_{\mathcal{Q}\cup\widetilde{\mathcal{U}}} = S_{\mathcal{Q}} - \widetilde{\mathcal{U}}$, such that at least one such pair $(\widetilde{\mathcal{U}}, \widetilde{\mu})$ satisfies the following properties:*

1. $|\widetilde{\mathcal{U}}| = |\mathcal{U}|$.
2. $\widetilde{\mathcal{U}} \cap \mathcal{R} = \emptyset$, *i.e., $\widetilde{\mathcal{U}}$ does not contain any center of $S_0 - \mathcal{Q}$ whose set of clients is completely assigned to a center of $\hat{OPT}$ in the assignment μ_O.*
3. *$\widetilde{\mu}$ satisfies:*
   (a) *For every center $c \in S_{\mathcal{Q}} - \widetilde{\mathcal{U}}$, we have $\widetilde{\mu}(p) = c$ for every $p \in S_{\mathcal{Q}}(c)$.*
   (b) *For a center $c \in \widetilde{\mathcal{U}}$, the clients in $S_{\mathcal{Q}}(c)$ are reassigned to a center $c' \in S - \mathcal{Q} - \widetilde{\mathcal{U}} - \mathcal{R}$, i.e., $\widetilde{\mu}(p) = c'$ for every $p \in S_{\mathcal{Q}}(c)$.*
   (c) *The cost increase of the reassignment $\widetilde{\mu}$ of clients previously assigned to $\widetilde{\mathcal{U}}$ compared to that of the reassignment μ_O of clients previously assigned to $\mathcal{U}$ is bounded by $O(\sqrt{\varepsilon}) \cdot opt$:*

$$\sum_{c \in \mathcal{U}\cup\widetilde{\mathcal{U}}} \sum_{p \in S_{\mathcal{Q}}(c)} \left(d^2(p, \widetilde{\mu}(p)) - d^2(p, \mu_O(p))\right) = O(\sqrt{\varepsilon}) \cdot opt.$$

In words, the above properties of $\widetilde{\mu}$ say that we maintain the assignment of clients whose closest center remains the same, and the clients associated to the removed centers $\widetilde{\mathcal{U}}$ are reassigned to centers not in $\mathcal{R}$ at no higher cost (up to $O(\sqrt{\varepsilon}) \cdot \text{opt}$) than the cost of the reassignment of $\mathcal{U}$ in the modified solution M_O.

9.9 Mixed Solutions after the Removal of $\widetilde{\mathcal{U}}$

We next focus on a pair $(\widetilde{\mathcal{U}}, \widetilde{\mu})$ that satisfies the properties in the claim of Lemma 9.14. Recall that $S_{\mathcal{Q}\cup\widetilde{\mathcal{U}}}$ consists of the centers $S' = S - \mathcal{Q} - \widetilde{\mathcal{U}}$ and the dummy centers Λ. Further recall that M_O consists of centers $S - S_0 = S - \mathcal{Q} - \mathcal{U} - \mathcal{R}$ and $\hat{\text{OPT}}$. M_D is the same set of centers except that $\hat{\text{OPT}}$ is replaced by Λ. Notice that some centers of M_O and M_D are now removed if $\widetilde{\mathcal{U}} - \mathcal{U} \neq \emptyset$. To take care of this, we modify these mixed solutions to obtain M'_O and M'_D.

We first define M'_O; the definition of M'_D is then very similar. The centers of M'_O are $S - \mathcal{Q} - \widetilde{\mathcal{U}} - \mathcal{R}$ and $\hat{\text{OPT}}$. Hence, the difference between M_O and M'_O is that in M_O we remove $\mathcal{U}$ from S and in M'_O we remove $\widetilde{\mathcal{U}}$. In other words, $M'_O = M_O \cup \mathcal{U} - \widetilde{\mathcal{U}}$. Similarly, we let $M'_D = M_D \cup \mathcal{U} - \widetilde{\mathcal{U}}$. We update the assignment μ_O of M_O to an assignment μ'_O of M'_O (and the assignment μ_D to μ'_D). Recall that μ_O assigns each client p to its closest center in $\hat{\text{OPT}}$ (if $p \in D^*_{\texttt{exp}}$) or its closest center in $S - S_0$ (if $p \in D - D^*_{\texttt{exp}}$) except for those clients that belong to a cluster $S_{\mathcal{Q}}(c)$ with $c \in \mathcal{U} \cup \mathcal{R}$. Indeed, the clients in $S_{\mathcal{Q}}(c)$ are all assigned to the center of $\hat{\text{OPT}}$ that is closest to c if $c \in \mathcal{R}$, and if $c \in \mathcal{U}$ they are all assigned to the center in $S - S_0$ that is closest to c. We also recall that μ_D is the same as μ_O except when $\mu_O(p) \in \hat{\text{OPT}}$ in which case $\mu_O(p)$ is replaced by its corresponding dummy center.

Definitions of μ'_O and μ'_D. For clients $p \in S_{\mathcal{Q}}(c)$ with $c \notin \mathcal{U} \cup \widetilde{\mathcal{U}} \cup \Lambda$, we define $\mu'_O(p) = \mu_O(p)$. For $p \in S_{\mathcal{Q}}(c)$ with $c \in \Lambda$, we let $\mu'_O(p)$ be the closest center in $\hat{\text{OPT}}$. Finally, for $p \in S_{\mathcal{Q}}(c)$ with $c \in \mathcal{U} \cup \widetilde{\mathcal{U}}$, we define $\mu'_O(p) = \widetilde{\mu}(p)$.

The assignment μ'_D is obtained in the same way from μ_D with the difference that we use Λ instead of $\hat{\text{OPT}}$: For clients $p \in S_{\mathcal{Q}}(c)$ with $c \notin \mathcal{U} \cup \widetilde{\mathcal{U}} \cup \Lambda$, we define $\mu'_D(p) = \mu_D(p)$. For $p \in S_{\mathcal{Q}}(c)$ with $c \in \Lambda$, we let $\mu'_D(p)$ be the closest center in Λ. Finally, for $p \in S_{\mathcal{Q}}(c)$ with $c \in \mathcal{U} \cup \widetilde{\mathcal{U}}$, we define $\mu'_D(p) = \widetilde{\mu}(p)$.

F: Isn't the latter center c itself?

This completes the definition of μ'_O and μ'_D. We remark that we have $\mu'_O(p) = \mu_O(p)$ and $\mu'_D(p) = \mu_D(p)$ for all $p \in S_{\mathcal{Q}}(c)$ with $c \notin \mathcal{U} \cup \widetilde{\mathcal{U}} \cup \Lambda$.

We continue by arguing that μ'_O is well-defined, i.e., that $\mu'_O(p) \in M'_O$ for every $p \in D$ (the proof for μ'_D is the same). This is immediate for a client $p \in S_{\mathcal{Q}}(c)$ with $c \in \Lambda$. For a client $p \in S_{\mathcal{Q}}(c)$ with $c \in \mathcal{U} \cup \widetilde{\mathcal{U}}$, it holds because, by Lemma 9.14, $\widetilde{\mu}(p)$ equals $c \in M'_O$ if $c \in \mathcal{U} - \widetilde{\mathcal{U}}$ and otherwise $\widetilde{\mu}(p) \in S - \mathcal{Q} - \widetilde{\mathcal{U}} - \mathcal{R} \subseteq M'_O$. It remains to verify that $\mu_O(p) \in M'_O$ for a client $p \in S_{\mathcal{Q}}(c)$ with $c \notin \mathcal{U} \cup \widetilde{\mathcal{U}} \cup \Lambda$. If $c \in \mathcal{R}$, we have $\mu_O(p) \in \hat{\text{OPT}} \subseteq M'_O$. Similarly, if $c \notin \mathcal{U} \cup \widetilde{\mathcal{U}} \cup \Lambda \cup \mathcal{R}$ and $p \in D^*_{\texttt{exp}}$ we have $\mu_O(p) \in \hat{\text{OPT}} \subseteq M'_O$. In the remaining case when $c \notin \mathcal{U} \cup \widetilde{\mathcal{U}} \cup \Lambda \cup \mathcal{R}$ and $p \in D - D^*_{\texttt{exp}}$, $\mu_O(p)$ equals p's closest center in $S - S_0$. As $p \in S_{\mathcal{Q}}(c)$, we thus have $\mu_O(p) = c \in S - (S_0 \cup \widetilde{\mathcal{U}}) \subseteq M'_O$.

The following upper bound on $\text{cost}(M'_O, \mu'_O) + \text{cost}(M'_D, \mu'_D)$ is a fairly immediate consequence ofăthe assumption that the cheap-removal process is successful (Property 3c of Lemma 9.14). Similarly to before, we say that the selection of $(W, \mathcal{B}, \mathcal{Q}, \widetilde{\mathcal{U}}, \widetilde{\mu})$ is successful, if the sample W selected in 9.3 is successful, the set of balls $\mathcal{B}$ from Subsection 9.4 is valid, $\mathcal{Q}$ selected in Subsection 9.6 satisfies the properties of Lemma 9.11, and $\widetilde{\mathcal{U}}, \widetilde{\mu}$ selected in this section satisfies the properties of Lemma 9.14. Again, there is an analogue of the next claim in [CCG+25], where however only the sum $\text{cost}(M'_O, \mu'_O) + \text{cost}(M'_D, \mu'_D)$ is upper bounded. We need to upper bound the two quantities separately.

Claim 9.15. *If $(W, \mathcal{B}, \mathcal{Q}, \widetilde{\mathcal{U}}, \widetilde{\mu})$ is successful,*

F: Separated the two bounds, the proof was already showing that

$$\textit{cost}(M'_O, \mu'_O) \leq \textit{cost}(M_O, \mu_O) + O(\sqrt{\varepsilon}) \cdot \textit{opt},$$

and

$$\textit{cost}(M'_D, \mu'_D) \leq \textit{cost}(M_D, \mu_D) + O(\sqrt{\varepsilon}) \cdot \textit{opt}.$$

Proof. For $x \in \{O, D\}$ the only differences between μ'_x and μ_x are clients in $S_{\mathcal{Q}}(c)$ with $c \in \Lambda \cup \mathcal{U} \cup \widetilde{\mathcal{U}}$. Consider first a client $p \in S_{\mathcal{Q}}(c)$ with $c \in \Lambda$. Then, as c is the closest center among $(S - \mathcal{Q}) \cup \Lambda$, which both contains $M_O - \hat{\text{OPT}}$ and $M'_O - \hat{\text{OPT}}$, we have that the closest

center to p in both M'_O and M_O is in $\hat{\text{OPT}}$ (because by the definition of dummy centers, $d^2(p, \hat{\text{OPT}}) \le d^2(p, \Lambda)$). Similarly, the closest center to p in both M'_D and M_D is in Λ. Hence, the definitions of μ'_O and μ'_D to assign p to its closest center in $\hat{\text{OPT}}$ and Λ, respectively, cannot increase the cost, i.e., $d^2(p, \mu'_O(p)) \le d^2(p, \mu_O(p))$ and $d^2(p, \mu'_D(p)) \le d^2(p, \mu_D(p))$ for such a client p.

Finally, for those clients $p \in S_{\mathcal{Q}}(c)$ with $c \in \mathcal{U} \cup \widetilde{\mathcal{U}}$ we have $\mu'_x = \widetilde{\mu}$ by definition and $d^2(p, \mu_O(p)) \le d^2(p, \mu_D(p))$ since $\mu_O(p) = \mu_D(p)$ unless $\mu_O(p) \in \hat{\text{OPT}}$ in which case $\mu_D(p)$ is the dummy center associated with $\mu_O(p)$ that can only be farther away from p than $\mu_O(p)$. Hence

$$\sum_{c \in \mathcal{U} \cup \widetilde{\mathcal{U}}} \sum_{p \in S_{\mathcal{Q}}(c)} \left(d^2(p, \widetilde{\mu}(p)) - d^2(p, \mu_D(p)) \right) \le \sum_{c \in \mathcal{U} \cup \widetilde{\mathcal{U}}} \sum_{p \in S_{\mathcal{Q}}(c)} \left(d^2(p, \widetilde{\mu}(p)) - d^2(p, \mu_O(p)) \right) = O(\sqrt{\varepsilon} \cdot \text{opt}),$$

where in the equality we used Property (c) of Lemma 9.14. □

To better understand $\text{cost}(M'_O, \mu'_O)$, let us consider the cost $d(p, \mu'_O(p))$ of a single client p:

- If $p \in S_{\mathcal{Q}}(c)$ with $c \notin \widetilde{\mathcal{U}} \cup \mathcal{R}$ then $\widetilde{\mu}(p) = c$ by Property 3a of Lemma 9.14. Moreover, $\mu'_O(p)$ assigns p either to a center in $\hat{\text{OPT}}$ or to one in $S - \mathcal{Q} - \widetilde{\mathcal{U}} - \mathcal{R}$ and we have $d^2(p, c) \le d^2(p, S - \mathcal{Q} - \widetilde{\mathcal{U}} - \mathcal{R})$. Hence, in either case, we have
$$d^2(p, \mu'_O(p)) \ge d^2(p, \{\widetilde{\mu}(p)\} \cup \hat{\text{OPT}}) = d^2(p, \{\widetilde{\mu}(p)\} \cup \hat{\text{OPT}} \cup \Lambda).$$
- If $p \in S_{\mathcal{Q}}(c)$ with $c \in \mathcal{R}$ then $\mu'_O(p) = \mu_O(p) = c^* \in \hat{\text{OPT}}$ where c^* is the center of $\hat{\text{OPT}}$ that is closest to c. So
$$d^2(p, \mu'_O(p)) = d^2(p, c^*).$$
- If $p \in S_{\mathcal{Q}}(c)$ with $c \in \widetilde{\mathcal{U}}$ then $\mu'_O(p) = \widetilde{\mu}(p)$ and so
$$d^2(p, \mu'_O(p)) \ge d^2(p, \{\widetilde{\mu}(p)\} \cup \hat{\text{OPT}} \cup \Lambda).$$

Similarly, we can analyze $d^2(p, \mu_D(p))$ by replacing $\hat{\text{OPT}}$ with the set Λ of dummy centers (and c^* by its associated dummy center). Summarizing, we have

$$\text{cost}(M'_O, \mu'_O) \ge \sum_{c \in S_{\mathcal{Q}} - \mathcal{R}} \sum_{p \in S_{\mathcal{Q}}(c)} d^2\left(p, \{c\} \cup \hat{\text{OPT}} \cup \Lambda\right) + \sum_{c \in \mathcal{R}} \min_{c' \in \hat{\text{OPT}}} \sum_{p \in S_{\mathcal{Q}}(c)} d^2\left(p, c'\right)$$

and

$$\text{cost}(M'_D, \mu'_D) \ge \sum_{c \in S_{\mathcal{Q}} - \mathcal{R}} \sum_{p \in S_{\mathcal{Q}}(c)} d^2\left(p, \{c\} \cup \Lambda\right) + \sum_{c \in \mathcal{R}} \min_{c' \in \Lambda} \sum_{p \in S_{\mathcal{Q}}(c)} d^2\left(p, c'\right)$$

Furthermore, by Properties 3a and 3b of Lemma 9.14, we have that, for every $c \in \mathcal{R}$, the set $S_{\mathcal{Q}}(c)$ of clients assigned to c in $S_{\mathcal{Q}}$ equals the set $\widetilde{\mu}^{-1}(c)$ of clients assigned by $\widetilde{\mu}$. So, if we let $\{C_c\}_{c \in S_{\mathcal{Q} \cup \widetilde{\mathcal{U}}}}$ be the partitioning of the set D of clients according to $\widetilde{\mu}$, i.e., $C_c = \{p \in D \mid \widetilde{\mu}(p) = c\}$, we can thus rewrite the above bounds on the cost as

F: I do not understand this sentence. Do we mean that for $p \in S_{\mathcal{Q}}(c)$, $\widetilde{\mu}(p) = c$?

$$\text{cost}(M'_O, \mu'_O) \ge \sum_{c \notin \mathcal{R}} \sum_{p \in C_c} d^2\left(p, \{c\} \cup \hat{\text{OPT}} \cup \Lambda\right) + \sum_{c \in \mathcal{R}} \min_{c' \in \hat{\text{OPT}}} \sum_{p \in C_c} d^2\left(p, c'\right) \tag{46}$$

and

$$\text{cost}(M'_D, \mu'_D) \geq \sum_{c \notin \mathcal{R}} \sum_{p \in C_c} d^2\left(p, \{c\} \cup \Lambda\right) + \sum_{c \in \mathcal{R}} \min_{c' \in \Lambda} \sum_{p \in C_c} d^2\left(p, c'\right) . \tag{47}$$

In the next section, we use these bounds to give a polynomial-time algorithm that outputs a solution S^* with k centers so that (see Lemma 9.16)

$$\begin{aligned}\text{cost}(S^*) &\leq (1 + 6\sqrt{\varepsilon}) \left((1 - \frac{1}{e}) \text{cost}(M'_O, \mu'_O) + \frac{1}{e} \text{cost}(M'_D, \mu'_D) \right) + O(\sqrt{\varepsilon}) \cdot \text{opt} \\ &\leq \sum_{p \in D} 4\text{OPT}_p + O(\sqrt{\varepsilon}) \cdot \text{opt}\end{aligned}$$

where the second inequality is by Lemma 9.12 and Claim 9.15. So the proof of Lemma 9.16 in the next subsection is the final step in the proof of Theorem 1.4 (see also Subsection 9.11 where we put everything together to prove Theorem 1.4).

9.10 Finding S^* via Submodular Optimization

In this section, we give a polynomial-time algorithm for finding the solution S^* by reducing the problem to maximizing a submodular function subject to a partition matroid constraint. Specifically, we prove the following lemma:

Lemma 9.16. *We can in polynomial-time find a clustering S^* with k centers of cost at most $(4 + O(\sqrt{\varepsilon})) \cdot opt$.*

Recall that at this point the algorithm calculated the local search solution S, sampled W, guessed $\mathcal{B}$, $\mathcal{Q}$, and $\widetilde{\mathcal{U}}$ with the assignment $\widetilde{\mu}$ of the clustering $S_{\mathcal{Q} \cup \widetilde{\mathcal{U}}}$. We assume that all these choices were successful guesses, i.e., that $(W, \mathcal{B}, \mathcal{Q}, \widetilde{\mathcal{U}}, \widetilde{\mu})$ is successful. Further recall the notation that we partition the clients into the clusters $\{C_c\}_{c \in S_{\mathcal{Q} \cup \widetilde{\mathcal{U}}}}$ where $C_c = \{p \in D : \widetilde{\mu}(p) = c\}$.

Our goal is to give a polynomial-time algorithm that finds approximations of the sets $\hat{\text{OPT}}$ and $\mathcal{R}$ that only have slightly worse cost. We start by defining the feasible set of candidates for $\hat{\text{OPT}}$ as a partition matroid. Recall that $\hat{\text{OPT}}$ contains one center from each ball in $\mathcal{B}$.

Definition of partition matroid $\mathcal{M}_{\mathcal{B}}$. We define the partition matroid that captures the constraint that we wish to open a center in each ball in $\mathcal{B}$. Let $F_{\mathcal{B}}$ be the (multi) subset of facility/center locations containing $B \cap F$ for each ball $B \in \mathcal{B}$. If a center c is in multiple balls in $\mathcal{B}$, then $F_{\mathcal{B}}$ contains one distinct copy of c for each ball. We let $F_{\mathcal{B}}(B) \subseteq F_{\mathcal{B}}$ denote the centers associated with $B \in \mathcal{B}$. These sets satisfy the following two properties:

- The sets $F_{\mathcal{B}}(B)$ partition $F_{\mathcal{B}}$.
- For $B \in \mathcal{B}$, $F_{\mathcal{B}}(B)$ contains (a copy of) every center in $F \cap B$.

The first property holds since we took a unique copy of each center for each ball, and they are thus disjoint: $F_{\mathcal{B}}(B) \cap F_{\mathcal{B}}(B') = \emptyset$ for distinct $B, B' \in \mathcal{B}$. Indeed, while it is not better for the cost to open multiple copies of a center, we make the copies to ensure the above two properties. This allows us to define the partition matroid $\mathcal{M}_{\mathcal{B}} = (F_{\mathcal{B}}, \mathcal{I})$ where

$$\mathcal{I} = \{X \subseteq F_{\mathcal{B}} : |X \cap F_{\mathcal{B}}(B)| \leq 1 \text{ for every } B \in \mathcal{B}\} .$$

Moreover, since there is exactly one ball in $\mathcal{B}$ for each center in $\hat{\text{OPT}}$, we have $\hat{\text{OPT}} \in \mathcal{I}$ (where we slightly abuse notation as we should take the copy of center $c^* \in \hat{\text{OPT}}$ that belongs to its associated ball).

Core, concentrated and hit clusters. Our goal is now to define a submodular function f so that we obtain a good approximation to $\hat{\text{OPT}}$ by maximizing f over the matroid constraint $\mathcal{M}_{\mathcal{B}}$. In particular, the domain of f is every subset of $F_{\mathcal{B}}$. However, we need some additional steps before defining f. In particular, we introduce the concept of the core of a cluster C_c and the notions of concentrated and hit clusters, which allow us to simplify the structure of centers in $\mathcal{R}$. For a cluster C_c, we define the *core* of C_c as

$$C_c^{\text{core}} = \left\{ p \in C_c : d^2(p,c) \leq \varepsilon \cdot \frac{\text{cost}(S)}{|\mathcal{R}| \cdot |C_c|} \right\}.$$

We further say that cluster C_c is *concentrated* if

$$|C_c^{\text{core}}| \geq (1-\varepsilon)|C_c|$$

and it is *hit* by a set X of centers if there is a point $p \in C_c^{\text{core}}$ such that

$$d^2(p,X) < d^2(p,c).$$

For shortness we will sometimes say that a center is concentrated (resp., hit), if the corresponding cluster is so.

Guessing the centers of $\mathcal{R}$ that are not concentrated. We further simplify the task of finding the set of centers $\mathcal{R}$ by guessing the centers of $\mathcal{R}$ that are not concentrated. Specifically, partition the set $\mathcal{R}$ into $\mathcal{R}_0$ and $\mathcal{R}_1$, where $\mathcal{R}_1$ contains those centers of $\mathcal{R}$ that are concentrated and $\mathcal{R}_0$ contains those that are not. We can correctly guess $\mathcal{R}_0$ in polynomial time since the following simple claim shows that it is a subset of the centers whose cluster costs at least $\varepsilon^2 \cdot \text{opt}/|\mathcal{R}|$, of which there are only $O(|\mathcal{R}|/\varepsilon^2)$ many, and so guessing $\mathcal{R}_0$ can thus be done in time $2^{O(|\mathcal{R}|/\varepsilon^2)}$.

Claim 9.17. *[CCG+25] Suppose that C_c is not concentrated. Then the cost of C_c is at least $\varepsilon^2 \cdot opt/|\mathcal{R}|$.*

Now let $\mathcal{P}$ be the potential centers that can be in $\mathcal{R}$. Specifically, we let $\mathcal{P}$ be the set that contains a center c in $S - \mathcal{Q} - \widetilde{\mathcal{U}}$ if C_c equals the set of clients in $S_{\mathcal{Q}}(c)$, where we recall that $S_{\mathcal{Q}}(c)$ is the set of clients that are closest to c in the clustering $S_{\mathcal{Q}}$. Notice that the algorithm has all the information $S, \mathcal{B}, \mathcal{Q}, \widetilde{\mathcal{U}}$ and $\widetilde{\mu}$ to calculate $\mathcal{P}$. Furthermore, by Lemma 9.14, we have that no clients from $\widetilde{\mathcal{U}}$ were reassigned by $\widetilde{\mu}$ to a center in $\mathcal{R}$. So for $c \in \mathcal{R}$ we have $C_c = S_{\mathcal{Q}}(c)$. Hence, $\mathcal{R} \subseteq \mathcal{P}$ and we can guess $\mathcal{R}_0$ as follows:

Guessing $\mathcal{R}_0$

1. Let $\mathcal{C}$ be the clusters of $\mathcal{P}$ whose cost is at least $\varepsilon^2 \cdot \text{opt}/|\mathcal{R}|$.
2. Output each subset of $\mathcal{C}$.

By the above claim and the definition of $\mathcal{P}$, one of the outputs is $\mathcal{R}_0$. Furthermore, the above guessing procedure outputs a family of polynomial many subsets. Indeed, we have $|\mathcal{R}| \leq \log(n)/\varepsilon^3$ by Lemma 9.3. Moreover, the total cost of the clusters in $\mathcal{P}$ is $O(\text{opt})$. To see this notice that the clusters corresponding to $\mathcal{P}$ is a subset of the clusters of $S_{\mathcal{Q}}$ and the cost of $S_{\mathcal{Q}}$ is at most $\text{cost}(S - S_0 \cup \Lambda)$ (since $S_{\mathcal{Q}} \supseteq S - S_0 \cup \Lambda$), which has cost at most $O(\text{opt})$ by Lemma 9.9. This implies that $|\mathcal{C}| = O(\log(n)/\varepsilon^5)$, so the total number of subsets is $n^{\varepsilon^{-O(1)}}$. The algorithm proceeds by trying all possible subsets in the output, and we analyze the algorithm when it takes the correct guess of $\mathcal{R}_0$.

Definition of the submodular function f. We first define another function g on the same domain as f, i.e., on all subsets X of $F_{\mathcal{B}}$. We will then define f by $f(X) = g(\emptyset) - g(X)$. Let the *closed cost* of a cluster C_c be defined as

$$\text{closedcost}_c(X) := \begin{cases} \sum_{p \in C_c} d^2(p, \{c\} \cup X \cup \Lambda) & \text{if } C_c \text{ is hit by } X, \\ \min_{c' \in X \cup \Lambda} \sum_{p \in C_c^{\text{core}}} d^2(p, c') + \sum_{p \in C_c - C_c^{\text{core}}} d^2(p, \{c\} \cup X \cup \Lambda) & \text{otherwise.} \end{cases}$$

Further, let $\mathcal{P}_1$ be the potential centers for $\mathcal{R}_1$: it contains each center $c \in \mathcal{P} - \mathcal{R}_0$ so that c is concentrated, i.e., $|C_c^{\text{core}}| \geq (1-\varepsilon)|C_c|$. We remark that the algorithm can calculate this set $\mathcal{P}_1$ as it only depends on $\mathcal{P}$, the guessed set $\mathcal{R}_0$, the value $\text{cost}(S)$, and the clusters C_c defined by $\widetilde{\mu}$. Moreover, by definition, we have $\mathcal{R}_1 \subseteq \mathcal{P}_1$. For a subset $X \subseteq F_{\mathcal{B}}$, we then define $g(X)$ to be the minimum value of

$$\sum_{c \notin \mathcal{R}_0 \cup \mathcal{R}_1'} \sum_{p \in C_c} d^2\left(p, \{c\} \cup X \cup \Lambda\right) + \sum_{c \in \mathcal{R}_0} \sum_{p \in C_c} d^2(p, X \cup \Lambda) + \sum_{c \in \mathcal{R}_1'} \text{closedcost}_c(X)$$

over all subsets $\mathcal{R}_1' \subseteq \mathcal{P}_1$ with $|\mathcal{R}_1'| = |\mathcal{R}_1| = |\mathcal{R}| - |\mathcal{R}_0|$. In words, over the best $\mathcal{R}_1'$, $g(X)$ is the cost of the solution obtained by removing the centers $\mathcal{R}_0 \cup \mathcal{R}_1'$ and assigning clients as follows:

- If $p \in C_c$ for a remaining center $c \notin \mathcal{R}_0 \cup \mathcal{R}_1'$ or $p \notin C_c^{\text{core}}$ with $c \in \mathcal{R}_1'$, p is assigned to its closest center in $\{c\} \cup X \cup \Lambda$.
- If $p \in C_c$ for a removed center $c \in \mathcal{R}_0$, p is assigned to its closest center in $X \cup \Lambda$.
- If $p \in C_c^{\text{core}}$ for a removed center $c \in \mathcal{R}_1'$, p is assigned to its closest center in $\{c\} \cup X \cup \Lambda$ if X hits C_c, and otherwise all clients in C_c^{core} are assigned to the same center $c' \in X \cup \Lambda$ that minimizes the cost.

We remark that this assignment is infeasible in the sense that it may assign clients to removed centers in $\mathcal{R}_1$. Nevertheless, we relate the values of $g(\emptyset)$ and $g(\hat{\text{OPT}})$ to $\text{cost}(M_D', \mu_D')$ and $\text{cost}(M_O', \mu_O')$, respectively; and, we show that given an $X \subseteq F_{\mathcal{B}}$ that is independent in $M_{\mathcal{B}}$, i.e., $X \in \mathcal{I}$, we can in polynomial-time output k centers S^* whose cost is at most $(1 + 6\sqrt{\varepsilon})g(X) + O(\sqrt{\varepsilon})\text{opt}$. Finally, the definition of g allows us to prove that f (defined by $f(X) = g(\emptyset) - g(X)$) is a non-negative monotone submodular function.

Lemma 9.18. *[CCG+25] We have that g and f satisfy the following properties:*

1. *We can evaluate $g(X)$ in polynomial time for every given $X \subseteq F_{\mathcal{B}}$, and we can thus evaluate $f(X)$ in polynomial time.*
2. *f is a non-negative monotone submodular function.*
3. *The value $g(\emptyset)$ is at most $\text{cost}(M_D', \mu_D')$.*
4. *The value $g(\hat{OPT})$ is at most $\text{cost}(M_O', \mu_O')$.*
5. *Given an $X \subseteq F_{\mathcal{B}}$ that is independent in $M_{\mathcal{B}}$, i.e., $X \in \mathcal{I}$, we can in polynomial-time output k centers S^* whose associated cost is at most $(1 + 6\sqrt{\varepsilon})g(X) + O(\sqrt{\varepsilon})\text{opt}$.*

This directly implies Lemma 9.16.

Proof of Lemma 9.16. By the first property of Lemma 9.18, we can evaluate f in polynomial time. Moreover, it is easy to see that we can answer independence queries in polynomial time for any partition matroid, particularly for $\mathcal{M}_{\mathcal{B}}$. We can thus apply Theorem 9.1 on f and $\mathcal{M}_{\mathcal{B}}$ with $\zeta = \varepsilon$ to find a solution $X \in \mathcal{I}$ such that

$$g(\emptyset) - g(X) = f(X) \geq (1 - 1/e - \varepsilon) f(\hat{\text{OPT}}) = (1 - 1/e - \varepsilon)(g(\emptyset) - g(\hat{\text{OPT}}))\,,$$

where we used that $\hat{\text{OPT}}$ is one feasible solution, i.e., $\hat{\text{OPT}} \in \mathcal{I}$. This in turn implies that

$$g(X) \leq (\frac{1}{e} + \varepsilon) g(\emptyset) + (1 - \frac{1}{e} - \varepsilon) g(\hat{\text{OPT}}).$$

Using the upper bounds on $g(\emptyset)$ and $g(\hat{\text{OPT}})$ of the above lemma we thus have found a set X such that

$$\begin{aligned} g(X) &\leq (\frac{1}{e} + \varepsilon)\text{cost}(M'_D, \mu'_D) + (1 - \frac{1}{e} - \varepsilon)\text{cost}(M'_O, \mu'_O) \\ &\overset{\text{Claim 9.15}}{\leq} \frac{1}{e}\text{cost}(M_D, \mu_D) + (1 - \frac{1}{e})\text{cost}(M_O, \mu_O) + O(\sqrt{\varepsilon})\text{opt}\,, \end{aligned}$$

where in the last inequality we also used that $\text{cost}(M_D, \mu_D)$ and $\text{cost}(M_O, \mu_O)$ are $O(1)\text{opt}$ by Lemma 9.12. Now using the last property of the above lemma we can output a solution S^* whose cost is at most

F: Just add the final 4 apx

$$\begin{aligned} &(1 + 6\sqrt{\varepsilon}) \left(\frac{1}{e}\text{cost}(M_D, \mu_D) + (1 - \frac{1}{e})\text{cost}(M_O, \mu_O) + O(\sqrt{\varepsilon})\text{opt} \right) + O(\sqrt{\varepsilon}) \cdot \text{opt} \\ \leq &\frac{1}{e}\text{cost}(M_D, \mu_D) + (1 - \frac{1}{e})\text{cost}(M_O, \mu_O) + O(\sqrt{\varepsilon}) \cdot \text{opt} \\ \overset{\text{Lem. 9.12}}{\leq} &\sum_{p \in D^*_{\text{exp}} - T_2} (1 + \frac{8}{e})\text{OPT}_p + \sum_{p \in D^*_{\text{exp}} \cap T_2} 4\text{OPT}_p \\ &+ \sum_{p \in (D - D^*_{\text{exp}}) - T_1} r(p) + \sum_{p \in (D - D^*_{\text{exp}}) \cap T_1} \frac{1}{4}(1 + \frac{8}{e}) r(p) + O(\sqrt{\varepsilon}) \cdot \text{opt} \\ \leq &\sum_{p \in D^*_{\text{exp}}} 4\text{OPT}_p + \sum_{p \in D - D^*_{\text{exp}}} r(p) + O(\sqrt{\varepsilon}) \cdot \text{opt} \overset{\text{Lem. 9.7}}{\leq} \sum_{p \in D} 4\text{OPT}_p + O(\sqrt{\varepsilon}) \cdot \text{opt} \end{aligned}$$

as required.

Finally, each of the above steps runs in polynomial time: the algorithm of Theorem 9.1 is polynomial time, and the last property of Lemma 9.18 used to obtain S^* is polynomial-time. Moreover, the number of guesses of $\mathcal{R}_0$ is at most $n^{\varepsilon^{-O(1)}}$, as argued after the description of that procedure. So we can, in polynomial time, try all possibilities and, among all solutions found (one for each guess of $\mathcal{R}_0$), return one that minimizes the cost and, in particular, has cost at most that of S^* (which was analyzed assuming the guess of $\mathcal{R}_0$ was correct). We thus have a polynomial time algorithm that returns a solution that satisfies the guarantee of the lemma. □

9.11 Putting Everything Together – Proof of Theorem 1.4

We now turn to proving Theorem 1.4. We first describe the algorithm we analyse.

A $(4+O(\sqrt{\varepsilon}))$-Approximation for k-Means on $\varepsilon/\log n$- Stable Instances k, D, F, d.

1. $S \leftarrow$ Local Search on (k, D, F, d), see Subsection 9.1
2. Let W be the set produced by the s^*-D^2-Sample process (Subsection 9.3) on S.
3. For each $\mathcal{B} \in \mathcal{L}_{\texttt{bal}}$ produced as in Lemma 9.8 on W (Subsection 9.4)
   (a) Compute the set of dummy centers Λ.
   (b) For each $Q \in \mathcal{L}_{\exp}$ produced as in Lemma 9.11 (Subsection 9.6):
      i. For each $\widetilde{\mathcal{U}} \in \mathcal{L}_{cheap}$, with the associated $\widetilde{\mu}$, produced as in Lemma 9.14 (Subsection 9.8):
      solve the Submodular Instance associated with $(W, \mathcal{B}, \mathcal{Q}, \widetilde{\mathcal{U}}, \widetilde{\mu})$ and compute the resulting solution $S_{(W,\mathcal{B},\mathcal{Q},\widetilde{\mathcal{U}},\widetilde{\mu})}$ for the k-Means problem (Subsection 9.10)
4. Output the solution $S_{(W,\mathcal{B},\mathcal{Q},\widetilde{\mathcal{U}},\widetilde{\mu})}$ of smallest cost.

Our approach consists of running the above algorithm $\log n$ times and take the minimum cost solution output. In the remaining, we argue that the above algorithm yields a $(4 + O(\sqrt{\varepsilon}))$-approximation with probability at least $(1-\varepsilon)(1-1/n)$. Theorem 1.4 then follows as the probability that all $\log n$ executions of the above algorithm would fail is at most $(1-(1-\varepsilon)(1-1/n))^{\log n} < 1/n$, and so with high probability we output a $(4 + O(\sqrt{\varepsilon}))$-approximation.

We first discuss the algorithm's running time. The local search algorithm runs in polynomial time. The submodular optimization step also runs in polynomial time by Lemma 9.16. The output of the s^*-D^2-Sample process (which is clearly a polynomial time procedure) is a subset of size s^*. By Lemma 9.8, we construct $\mathcal{L}_{\texttt{bal}}$ in polynomial time, and $|\mathcal{L}_{\texttt{bal}}| \leq n^{\varepsilon^{-O(1)}}$. Similarly, by Lemma 9.11, we construct $\mathcal{L}_{\exp}$ in polynomial time and $|\mathcal{L}_{\exp}| \leq n^{\varepsilon^{-O(1)}}$. Finally, by Lemma 9.14, we construct in polynomial-time $n^{\varepsilon^{-O(1)}}$ candidate pairs $\widetilde{\mathcal{U}}, \widetilde{\mu}$. It follows that we solve $n^{\varepsilon^{-O(1)}}$ submodular function optimization problems, each in polynomial time, and thus the total running time is polynomial.

We turn to proving the approximation guarantee. We aim to show that there exists a $(W, \mathcal{B}, \mathcal{Q}, \widetilde{\mathcal{U}}, \widetilde{\mu})$ that is successful with probability at least $(1-\varepsilon/2)$. Assuming this, the $(4+O(\sqrt{\varepsilon}))$-approximation follows by Lemma 9.16.

Our algorithm runs the local search algorithm and thus provides us with a 26 approximation S that satisfies Lemma 9.4. Our algorithm next applies the s^*-D^2-Sample process and by Lemma 9.5 finds a successful W with probability at least $1-\varepsilon$. Condition on having a successful W, we then have that Lemma 9.6 holds, and so does Lemma 9.7 by combining Lemma 9.4 and Lemma 9.6.

Next, Lemma 9.8 implies that one set of balls $\mathcal{B}$ is a valid set of balls. This set of balls induces a set of dummy centers Λ. From there, by Lemma 9.11 we have that with probability at least $1-1/n$, the procedure from Lemma 9.11 produces a set of centers $\mathcal{Q}$ such that

1. $\mathcal{Q} \subseteq S_0$, and
2. The total cost in solution $S - Q \cup \Lambda$ of the clusters in $S_0 - \mathcal{Q}$ is at most $\varepsilon \cdot \text{opt}$.

and so we have obtained a successful $(W, \mathcal{B}, \mathcal{Q})$ and Lemma 9.12 applies. Condition on the event that we obtain a successful $(W, \mathcal{B}, \mathcal{Q})$, then, the procedure from Lemma 9.14 outputs a pair $(\widetilde{\mathcal{U}}, \widetilde{\mu})$ satisfying the properties in the claim of the lemma. We thus have a $(W, \mathcal{B}, \mathcal{Q}, \widetilde{\mathcal{U}}, \widetilde{\mu})$ such that W is successful, the set of balls $\mathcal{B}$ from Subsection 9.4 is valid, $\mathcal{Q}$ selected in Subsection 9.6 satisfies the properties of Lemma 9.11, and $\widetilde{\mathcal{U}}, \widetilde{\mu}$ selected in this section satisfies the properties of Lemma 9.14 and so $(W, \mathcal{B}, \mathcal{Q}, \widetilde{\mathcal{U}}, \widetilde{\mu})$ is successful. Finally, note that the only probabilistic steps were the sampling of W and of $\mathcal{Q}$, and both steps are successful with probability at least $(1-\varepsilon) \cdot (1-1/n)$, as required.

10 $(2 + O(\sqrt{\varepsilon}))$-Approximation for $(\frac{\zeta}{\log n})$-Stable Euclidean Instances

In this section, we prove Theorem 1.5. Recall that our goal is to compute a $2 + O(\sqrt{\varepsilon})$ approximate solution for $O(\zeta/\log n)$ stable instances. This improves on the $4 + O(\sqrt{\varepsilon})$ approximation in [CCG+26], that however works in the more general metric setting, under the only restriction that the client locations are valid facility locations (while our improvement only works for the Euclidean case). Our algorithm is the same as in [CCG+26], and our analysis is very similar too: in the following we will only focus on the differences in the analysis.

Notation and Preliminaries. To simplify the formulas, we let ε be the minimum of ε and ζ in the claim of Theorem 1.5 and further assume that $\varepsilon < 1/12$ is sufficiently small. With this notation, we present a $(2+O(\sqrt{\varepsilon}))$-approximation algorithm for $\varepsilon/\log(n)$-stable instances. This implies the above theorem as any $\zeta/\log(n)$-stable instance is also $\zeta'/\log(n)$-stable with $\zeta' \leq \zeta$.

Recall that, by known reductions, we can assume that we are given a set F of tentative centers, with $D \subseteq F$, and we have to choose centers from that set. Recall also that non-zero distances are between 1 and n^2/ε^2 according to Lemma 3.3. We also assume w.l.o.g. that the optimal cost is positive (hence at least 1).

Given a cluster C^* with center $c^* \in F$, in the analysis it is convenient to consider the actual center of mass $\mu(C^*)$ of C^*. Obviously $\sum_{p \in C^*} d^2(\mu(C^*), p) \leq \sum_{p \in C^*} d^2(c^*, p)$. We will also need the following standard fact.

Lemma 10.1. *For any $C^* \subseteq D$ and for any $c, c^* \in \Re^t$,*

$$\sum_{p \in C^*} d^2(c,p) \leq |C^*| d^2(c, \mu(C)) + \sum_{p \in C^*} d^2(\mu(C^*), p) \leq |C^*| d^2(c, \mu(C^*)) + \sum_{p \in C^*} d^2(c^*, p).$$

Given a set of centers $A \subseteq F$ (not necessarily of cardinality k), we let $A_p = d^2(p, A)$ denote the cost associated to p in the clustering induced by A, and $\mathrm{cost}(A) := \sum_{p \in D} A_p$ be the total cost of A. The cluster in A associated with $c \in A$ is denoted by $A(c)$. Sometimes we will consider a (possibly suboptimal) assignment $\mu : D \to A$ of clients to centers in A, and let $\mathrm{cost}(A, \mu) := \sum_{p \in D} d^2(p, \mu(p))$ be the corresponding cost. Let OPT be a fixed optimum solution under the restriction $\mathrm{OPT} \subseteq F$. Let also $\mathrm{opt} = \mathrm{cost}(\mathrm{OPT})$ be its cost.

Overview of our approach. Our algorithm consists of a few steps which are discussed in the following subsections. The first step (see Section 10.1) is to compute a candidate solution S via standard local search. We will show that the points in almost all the clusters in OPT (excluding a logarithmic size set) pay in S roughly the same cost as in OPT. We call the

centers OPT_{pure} of such clusters *pure*. More precisely, we will show that all the points in $OPT(c^*)$ with $c^* \in \mathrm{OPT}_{pure}$ can be reassigned to the *same* center $c' \in S$ without increasing too much their cost.

For each remaining *impure* center c, we identify a core set of clients in $\mathrm{OPT}(c)$, the *leaders*, which are *sufficiently close* to c. We distinguish the impure clusters depending on whether the cost in S of the leaders is small (*core-cheap*) or large (*core-expensive*). The next step (see Section 10.2) is to identify one leader in most of the core-expensive clusters of OPT. Let Λ be the set of such leaders, and OPT_{exp} be the core-expensive centers for which one leader was discovered. One can reassign all the clients $\mathrm{OPT}(c^*)$ with $c^* \in \mathrm{OPT}_{exp}$ to the corresponding leader while increasing their cost at most by a factor (roughly) 2. Let OPT_{cheap} be the remaining impure centers of OPT. We show that we can reassign all the clients $OPT(c^*)$ with $c^* \in \mathrm{OPT}_{cheap}$ to the same center in S while increasing their cost, again, by at most a factor (roughly) 2. The latter two mentioned factors 2 are replaced by 4 in the analysis in [CCG+26].

Consider next the solution that contains the leaders Λ plus the centers in S to which the points in the clusters of $\mathrm{OPT}_{pure} \cup \mathrm{OPT}_{cheap}$ are reassigned (which are at most $|\mathrm{OPT}_{pure}| + |\mathrm{OPT}_{cheap}|$ many). Notice that this is a feasible solution since $|\Lambda| = |\mathrm{OPT}_{exp}|$. Furthermore, it is (roughly) 2-approximate by the previous discussion. Our final goal is to identify such a solution, or some good enough approximation of it. More precisely, we start with the solution $\Lambda \cup S$ (which is infeasible), and remove a set $S'_{disc} \subseteq S$ of $|\Lambda|$ many carefully chosen centers. This removal is done in two steps. First (see Section 10.3), we remove a set $\mathcal{Q} \subseteq S$ of centers which are *expensive* in S. Then (see Section 10.4), we remove other $\widetilde{\mathcal{U}}$ centers from S, with $|\widetilde{\mathcal{U}}| = |\Lambda| - |\mathcal{Q}|$.

10.1 Locally Optimal Solution and Pure Clusters

Let S be a locally optimal solution output by the following standard local search algorithm. Here, for technical reasons and differently from [CCGG98], we use a truncated version of local search: this way we do not need to assume that distances are integers.

localSearch()

1: $S \leftarrow$ Arbitrary solution for k-Means
2: **while** There exists a solution S' such that $|S \Delta S'| \leq 2$ and $\mathrm{cost}(S') \leq (1-\varepsilon)\mathrm{cost}(S)$ **do**
3: $\quad S \leftarrow S'$
4: **return** S

We remark that the running time of the above procedure is polynomial. It is standard fact that local search achieves a 26 approximation for k-Means, see [GT08] (as we will see, the specific constant in not very relevant for us).

Lemma 10.2. *[GT08] The procedure localSearch() runs in polynomial time and returns a feasible solution S with cost$(S) \leq 26 \cdot opt$.*

We next fix S, and recall that we also fixed a reference optimal solution OPT. We identify in OPT a set of pure clusters as follows. We say that a cluster C^* of OPT is *pure* (w.r.t. S) if there exists a cluster C' of S such that $|C' \Delta C^*| \leq 3\varepsilon \min(|C^*|, |C'|)$. In which case we also say that C' is pure w.r.t. OPT. We say that C' and C^* are associated. By $\mathrm{OPT}_{pure} \subseteq \mathrm{OPT}$ we denote the set of the centers of pure clusters (that we also call *pure*). The centers (resp, clusters) which are not pure are *impure*. The impure centers of OPT are denoted by OPT_{imp}. We use

a similar notation for the centers and clusters of S. The following lemma follows from the fact that the input is β-stable for $\beta := \varepsilon/\log n$ and that the solution S is a 26-approximation by Lemma 10.2. Given a client or center a and a distance r, we let $B(a,r)$ be the set of clients and centers at distance at most r from a. We will use the following Lemma from [CCG+25] (for technical reasons the original proof uses 25 as the approximation ratio of the local search solution, however their proof works exactly in the same way with 26).

Lemma 10.3. *[CCG+25]* $|OPT_{imp}| \leq \frac{\log n}{\varepsilon^3}$.

We next define a replacement cost $r(p)$ for each client $p \in D$. Let C^* be a pure cluster of OPT and C' be the associated cluster of S. Let c^* be the center of C^* and c' be the center of C'. We define $t(c^*) := c'$. For any client $p \in C^*$, let $r(p) := d^2(p,c') = d^2(p,t(c^*))$. For any impure cluster C^* with center c^*, we let $t(c^*)$ be the center of S closest to $\mu(C^*)$, and define $r(p) = d^2(p,t(c^*))$ for each $p \in C^*$.

F: Was c^*. Critical change to address issue 16

The following lemma is the only one that requires that the solution S is obtained via local search (the proof of the previous lemma only used that it was a constant-factor approximation). In words, it says that S approximates the connection cost of clients belonging to pure clusters of the optimal solution almost perfectly. We will use the following Lemma from [CCG+25].

Lemma 10.4. *[CCG+25]* $\sum_{c^* \in OPT_{pure}} \sum_{p \in OPT(c^*)} r(p) \leq \sum_{c^* \in OPT_{pure}} \sum_{p \in OPT(c^*)} OPT_p + O(\sqrt{\varepsilon}) \cdot opt$.

AS: changed the label here. check. I noticed that we never cite this lemma later. The reason is that it is used in the proof of some other lemma which is omitted here. We can leave this as it is, or remove the lemma and the previous paragraph

10.2 Leaders' Sampling of Leader-Expensive Clusters of OPT

Let $\mathrm{OPT}_{imp} = \mathrm{OPT} - \mathrm{OPT}_{pure}$ be the *impure* centers of OPT. We next partition OPT_{imp} according to the following definition. Consider any $c^* \in \mathrm{OPT}_{imp}$ and set $C^* = \mathrm{OPT}(c^*)$. We let $\mathrm{avg}(c^*)$ be the squared distance from $\mu(C^*)$ to the $\varepsilon|C^*|$-th closest client of C^* to $\mu(C^*)$. We further let $\mathrm{OPT}_{lead}(c^*)$ be the set of clients of C^* at a squared distance at most $\mathrm{avg}(c^*)$ from $\mu(C^*)$: we call such clients *leaders* of the cluster. So $\mathrm{avg}(c^*)$ is the *maximum squared distance* from a leader to $\mu(C^*)$. Also note that we have, $|\mathrm{OPT}_{lead}(c^*)| \geq \varepsilon|C^*|$. We thus have that $\mathrm{avg}(c^*)$ is roughly upper bounded by the average optimal cost of C^*, namely

F: Replaced c^* with $\mu(c^*)$ to address issue 16

$$\mathrm{avg}(c^*) \leq \frac{1}{(1-\varepsilon)|C^*|} \sum_{p \in C^*} d^2(p,\mu(C^*)) \leq \frac{1}{(1-\varepsilon)|C^*|} \sum_{p \in C^*} \mathrm{OPT}_p. \tag{48}$$

F: Put in equation form and modified

At the same time $\mathrm{OPT}_{lead}(c^*)$ contains a constant fraction of the clients in $\mathrm{OPT}(c^*)$. We say that an impure center $c^* \in \mathrm{OPT}_{imp}$ is *leader-cheap* if the total cost in S of the clients in $\mathrm{OPT}_{lead}(c^*)$ is less than $\varepsilon^5 \mathrm{opt}/\log n$, i.e., $\sum_{p \in \mathrm{OPT}_{\mathrm{avg}(c^*)}} S_p \leq \frac{\varepsilon^5}{\log n}\mathrm{opt}$. The remaining impure centers are called leader-expensive. We name similarly the respective clusters. Let OPT_{lcheap} and OPT_{lexp} be the leader-cheap and leader-expensive centers, resp. The next lemma shows that leader-cheap clusters have a small replacement cost. There is a similar lemma in [CCG+26] where the factor 2 is replaced by 4.

Lemma 10.5. $\sum_{c^* \in OPT_{lcheap}} \sum_{p \in OPT(c^*)} r(p) \leq \sum_{c^* \in OPT_{lcheap}} \sum_{p \in OPT(c^*)} 2\, OPT_p + O(\sqrt{\varepsilon})\, opt$.

Proof. Consider any $c^* \in \mathrm{OPT}_{lcheap}$, and let $C^* := \mathrm{OPT}(c^*)$ and $C^*_{lead} := \mathrm{OPT}_{lead}(c^*)$. One has

F: In a few placed replaced c^* with $\mu(C^*)$

$$\sum_{p \in C^*} d^2(p,t(c^*)) \overset{\text{Lem. } 10.1}{\leq} |C^*| d^2(\mu(C^*),t(c^*)) + \sum_{p \in C^*} \mathrm{OPT}_p.$$

Consider any $\ell \in C^*_{lead}$ and let $c(\ell)$ be the center closest to ℓ in S. Using Lemma 3.4 with $\gamma = 1 + \sqrt{\varepsilon}$,

$$d^2(\mu(C^*), t(c^*)) \le d^2(\mu(C^*), c(\ell)) \overset{\text{Lem. 3.4}}{\le} (1+\sqrt{\varepsilon}) d^2(\mu(C^*), \ell) + \frac{2}{\sqrt{\varepsilon}} d^2(\ell, c(\ell))$$
$$\le (1+\sqrt{\varepsilon}) \text{avg}(c^*) + \frac{2}{\sqrt{\varepsilon}} S_\ell.$$

Averaging over $\ell \in C^*_{lead}$,

$$d^2(\mu(C^*), t(c^*)) \le (1+\sqrt{\varepsilon}) avg(c^*) + \frac{2}{\sqrt{\varepsilon}|C^*_{lead}|} \sum_{\ell \in C^*_{lead}} S_\ell \le (1+\sqrt{\varepsilon}) avg(c^*) + \frac{2}{\varepsilon^{1.5}|C^*|} \frac{\varepsilon^5}{\log n} \text{opt},$$

where in the last inequality we used that $|C^*_{lead}| \ge \varepsilon |C^*|$ and that C^* is leader-cheap. Recall that $\text{avg}(c^*) \le \frac{1}{(1-\varepsilon)|C^*|} \sum_{p \in C^*} \text{OPT}_p$ by (48). Altogether

$$\sum_{p \in C^*} d^2(p, t(c^*)) \le \sum_{p \in C^*} \text{OPT}_p + \frac{1+\sqrt{\varepsilon}}{1-\varepsilon} \sum_{p \in C^*} \text{OPT}_p + \frac{2\varepsilon^{3.5}}{\log n} \text{opt}.$$

The claim follows by summing over $c^* \in \text{OPT}_{lcheap}$ and recalling that $|\text{OPT}_{lcheap}| \le |\text{OPT}_{imp}| \le \frac{\log n}{\varepsilon^3}$ by Lemma 10.3. □

Our next goal is to identify one leader for each leader-expensive center. We will actually miss each one of the latter centers with some small probability, however this is not a problem as we will see. The idea is to sample each client p with probability proportional to its cost S_p for a sufficiently large number of times: this way it is unlikely to not hit each set $\text{OPT}_{lead}(c^*)$ of a leader-expensive center c^*. Then one extracts a proper set of leaders by brute force. Here we critically exploit the fact that there is only a logarithmic number of impure centers (by Lemma 10.3), hence of leader-expensive centers. We will use the following result in [CCG+26] that works when S is $O(1)$ approximate.

Lemma 10.6. *[CCG+26] There is a polynomial-time procedure to compute a collection $\mathcal{L}_{lead}$ of subsets of clients such that, with probability at least $1-\varepsilon$, at least one $\Lambda \in \mathcal{L}_{lead}$ is* valid, *i.e., it satisfies the following property. Let $OPT_{exp} \subseteq OPT_{lexp}$ be the centers with exactly one leader in Λ and $OPT_{miss} = OPT_{lexp} - OPT_{exp}$. Then*

$$\sum_{c^* \in OPT_{miss}} \sum_{p \in OPT(c^*)} r(p) \le 4\varepsilon \, opt.$$

We next assume that the event in the above lemma happens, and focus on the corresponding value of Λ. For $c^* \in \text{OPT}_{exp}$, we let $lead(c^*)$ be the leader in $\text{OPT}(c^*) \cap \Lambda$. Let us define OPT_{exp} and OPT_{miss} as in the above lemma. Let also $\text{OPT}_{cheap} = \text{OPT}_{lcheap} \cup \text{OPT}_{miss}$. We call *expensive* and *cheap* the centers in OPT_{exp} and OPT_{cheap}, resp. Similarly for the respective clusters. The next lemma upper bounds the distance of the points in expensive clusters to the corresponding leaders. There is an analogous lemma in [CCG+26] where the 2 is replaced by a 4.

Lemma 10.7. *Assume that Λ is valid. One has*

$$\sum_{c^* \in OPT_{exp}} \sum_{p \in OPT(c^*)} d^2(p, \Lambda) \le \sum_{c^* \in OPT_{exp}} \sum_{p \in OPT(c^*)} 2 OPT_p + O(\varepsilon) opt$$

.

Proof. Consider any $c^* \in \mathrm{OPT}_{exp}$. Let $C^* := \mathrm{OPT}(c^*)$ and $\ell = lead(c^*) \in \mathrm{OPT}_{lead}(c^*) \cap \Lambda$. Notice that ℓ is well-defined by the definition of OPT_{exp}. Then

$$\sum_{p \in C^*} d^2(p, \Lambda) \overset{\text{Lem. 10.1}}{\leq} |C^*| d^2(\mu(C^*), \ell) + \sum_{p \in C^*} \mathrm{OPT}_p \leq |C^*| \mathrm{avg}(c^*) + \sum_{p \in C^*} \mathrm{OPT}_p$$
$$\overset{(48)}{\leq} (1 + O(\varepsilon)) \sum_{p \in C^*} \mathrm{OPT}_p.$$

□

The next lemma shows that there exists a choice of $|\mathrm{OPT}_{exp}|$ many centers $S_{disc} \subseteq S$, such that, replacing S_{disc} with Λ in S leads to a roughly 2 approximate solution. Let $D_{exp} = \cup_{c^* \in \mathrm{OPT}_{exp}} \mathrm{OPT}(c^*)$ and $D_{cheap} = D - D_{exp}$. The proof of the following lemma is almost identical to the one of an analogous lemma in [CCG+26] (where the 2 is replaced by a 4).

Lemma 10.8. *Suppose that Λ is valid. Then there exists a set $S_{disc} \subseteq S$ that satisfies the following properties:*

1. $|S_{disc}| = |OPT_{exp}| = |\Lambda| \leq \frac{\log n}{\varepsilon^3}$;
2. *for every $c \in S_{disc}$, there is no $c^* \in OPT - OPT_{exp}$ such that $t(c^*) = c$;*
3. $cost(S - S_{disc} \cup \Lambda) \leq 2 \sum_{p \in D_{exp}} OPT_p + \sum_{p \in D_{cheap}} r(p) \leq (2 + O(\sqrt{\varepsilon}))opt.$

Proof. By definition, the centers in OPT_{exp} are not pure, hence $|\Lambda| = |\mathrm{OPT}_{exp}| \leq \frac{\log n}{\varepsilon^3}$ by Lemma 10.3. For each center $c^* \in \mathrm{OPT}_{pure} \cup \mathrm{OPT}_{cheap}$, there is one associated center $t(c^*)$ in S, hence the total number of the latter centers $S' \subseteq S$ is at most $|\mathrm{OPT}_{pure}| + |\mathrm{OPT}_{cheap}| = k - |\mathrm{OPT}_{exp}|$. Let us set S_{disc} to any subset of $|\mathrm{OPT}_{exp}|$ centers in $S - S'$. The first two bullets follow.

Let us turn to the last bullet. By Lemma 10.7

$$\sum_{p \in D_{exp}} d^2(p, S \cup \Lambda - S_{disc}) \leq \sum_{p \in D_{exp}} d^2(p, \Lambda) \leq \sum_{p \in D_{exp}} 2\mathrm{OPT}_p + O(\varepsilon)\mathrm{opt}.$$

From the second bullet, for each client p in a cluster of the optimum solution whose center is $c^* \in \mathrm{OPT}_{cheap}$, we have that $t(c^*) \in S - S_{disc}$ and so its cost is at most $r(p)$. Thus, by Lemmas 10.5 and 10.6,

$$\sum_{p \in D_{cheap}} d^2(p, S \cup \Lambda - S_{disc}) \leq \sum_{p \in D_{cheap}} r(p) \leq \sum_{p \in D_{cheap}} 2\mathrm{OPT}_p + O(\sqrt{\varepsilon})\mathrm{opt}.$$

The last bullet follows. □

10.3 Removal of Expensive Centers of S_{disc}

Let us start with the tentative solution $S \cup \Lambda$ (which is infeasible since it contains more than k centers). Our goal is to remove a proper subset S'_{disc} of $|\Lambda|$ many centers from S, so that $S \cup \Lambda - S'_{disc}$, which is a feasible solution, is roughly 2-approximate. Ideally we would like to choose $S'_{disc} = S_{disc}$, so that Lemma 10.8 directly gives what we need. We will rather find

some S'_{disc} which is comparably good. For a subset $A \subseteq S$, let $S_A := S \cup \Lambda - A$. As usual, we let $S_A(c)$ be the cluster in S_A associated with the center $c \in S_A$.

Consider any cluster C of $S \cup \Lambda$ with center $c \in S_{disc}$. For reasons that will become clearer later, if we remove c (which is then added to S'_{disc}), we would like to assign all the clients in C to the same center c' in $S \cup \Lambda - \{c\}$. Similarly for the later steps. However, this might be too expensive when the cost of C in $S \cup \Lambda$ is large. Our next goal is to identify a subset $\mathcal{Q} \subseteq S_{disc}$ of centers such that, after removing $\mathcal{Q}$ from $S \cup \Lambda$, we enforce in some sense the above property. More precisely, we desire that the cost of the clusters with center in $S_{disc} - \mathcal{Q}$ in the solution $S_{\mathcal{Q}} := S \cup \Lambda - \mathcal{Q}$ is very small compared with opt. This is done via the following lemma in [CCGG98] that is based on the fact that $\text{cost}(\Lambda \cup S - S_{disc}) \leq O(1) \cdot \text{opt}$.

Lemma 10.9. *[CCG+26] There is a procedure that runs in time $n^{1/\varepsilon^{O(1)}}$ and produces a collection $\mathcal{L}_{exp}$ of at most $n^{1/\varepsilon^{O(1)}}$ subsets of S such that with probability at least $1 - 1/n$ there exists $\mathcal{Q} \in \mathcal{L}_{exp}$ satisfying*

1. *$\mathcal{Q} \subseteq S_{disc}$, and*
2. *The total cost in $S_{\mathcal{Q}}$ of the clusters with centers in $\mathcal{U} := S_{disc} - \mathcal{Q}$ is at most $\varepsilon \cdot opt$, namely $\sum_{c \in \mathcal{U}} \sum_{p \in S_{\mathcal{Q}}(c)} d^2(p, c) \leq \varepsilon \cdot opt$.*

10.4 Removal of Cheap Centers from S

Our algorithm considers each choice of $\mathcal{Q} \in \mathcal{L}_{exp}$. We next assume that there exists at least one such Q that satisfies the properties in Lemma 10.9 (event which happens with high probability), and focus on the execution of the algorithm corresponding to the latter $\mathcal{Q}$. Consider the (possibly infeasible) solution $S_{\mathcal{Q}}$. Our target reference solution is $S_{S_{disc}}$, which is roughly 2-approximate. Let $\mathcal{U} := S_{disc} - \mathcal{Q}$. Ideally we would like to remove $\mathcal{U}$ from $S_{\mathcal{Q}}$, hence getting the desired solution $S_{\mathcal{Q} \cup \mathcal{U}} = S_{S_{disc}}$. We will rather remove a subset $\mathcal{U}'$ of the same cardinality, whose removal leads to a solution of cost only marginally larger than the cost of $S_{S_{disc}}$. In order to simplify our argument, for each candidate set of centers $\mathcal{U}' \subseteq S - \mathcal{Q}$ with $|\mathcal{U}'| = |\mathcal{U}|$, we will restrict our attention to suboptimal assignments $\mu_{\mathcal{U}'}$ of points to centers in $S_{\mathcal{Q} \cup \mathcal{U}'}$ such that, for each $c \in \mathcal{U}'$, all the points $p \in S_{\mathcal{Q}}(c)$ are assigned to the closest center $t(c, \mathcal{U}') \in S_{\mathcal{Q} \cup \mathcal{U}'}$ to c. The remaining points are assigned by $\mu_{\mathcal{U}'}$ to the same center as in $S_{\mathcal{Q}}$. We call *consistent* an assignment $\mu_{\mathcal{U}'}$ of this type, and we denote the *consistent cost* of the corresponding solution by

$$cost'(S_{\mathcal{Q} \cup \mathcal{U}'}) = \sum_{p \in D} d^2(p, \mu_{\mathcal{U}'}(p)).$$

Obviously $cost'(S_{\mathcal{Q} \cup \mathcal{U}'})$ is an upper bound on the actual cost $cost(S_{\mathcal{Q} \cup \mathcal{U}'})$ of the solution $S_{\mathcal{Q} \cup \mathcal{U}'}$. The next lemma shows that, for $\mathcal{U}' = \mathcal{U}$, the consistent cost is small enough. The first inequality in the following lemma follows directly from an analogous lemma in [CCG+26], while the second from Lemma 10.8.3.

Lemma 10.10. $cost'(S_{\mathcal{Q} \cup \mathcal{U}}) \leq cost(S_{\mathcal{Q} \cup \mathcal{U}}) + O(\sqrt{\varepsilon}) opt \leq (2 + O(\sqrt{\varepsilon})) opt.$

Our next goal is to compute a set $\widetilde{\mathcal{U}}$ of cardinality $|\mathcal{U}|$ such that

$$cost'(S_{\mathcal{Q} \cup \widetilde{\mathcal{U}}}) \leq cost'(S_{\mathcal{Q} \cup \mathcal{U}}) + O(\sqrt{\varepsilon})\text{opt}.$$

This is done via the following lemma from [CCG+26].

Lemma 10.11. *[CCG+26] Assume that Λ and $\mathcal{Q}$ satisfy the properties mentioned before. Then there is a polynomial-time procedure that computes a polynomial size collection $\mathcal{L}_{cheap}$ of subsets of $S - \mathcal{Q}$, each of size $|\Lambda| - |\mathcal{Q}|$, such that at least one $\widetilde{\mathcal{U}} \in \mathcal{L}_{cheap}$ satisfies $cost'(S_{\mathcal{Q}\cup\widetilde{\mathcal{U}}}) \le cost'(S_{\mathcal{Q}\cup\mathcal{U}}) + O(\sqrt{\varepsilon})opt$.*

10.5 Putting Everything Together – Proof of Theorem 1.5

We next summarize the overall algorithm:

apxStableEuclideankMeans()

1: $\mathcal{L}_{apx} \leftarrow \emptyset$
2: $S \leftarrow localSearch()$ (see Subsection 10.1)
3: **for** $\frac{1}{2\varepsilon} \ln n$ many times **do**
4: Compute $\mathcal{L}_{lead}$ according to Lemma 10.6 (see Subsection 10.2)
5: **for** all $\Lambda \in \mathcal{L}_{lead}$ **do**
6: Compute $\mathcal{L}_{exp}$ according to Lemma 10.9 (see Subsection 10.3)
7: **for** all $\mathcal{Q} \in \mathcal{L}_{exp}$ **do**
8: Compute $\mathcal{L}_{cheap}$ according to Lemma 10.11 (see Subsection 10.4)
9: **for** all $\widetilde{\mathcal{U}} \in \mathcal{L}_{cheap}$ **do**
10: $\mathcal{L}_{apx} \leftarrow \mathcal{L}_{axp} \cup \{S \cup \Lambda - \mathcal{Q} - \widetilde{\mathcal{U}}\}$
11: **return** the cheapest solution $APX \in \mathcal{L}_{apx}$

We are now ready to prove Theorem 1.5.

Proof of Theorem 1.5. Consider the algorithm *apxStableEuclideankMeans*(). Its running time is polynomial since the called procedures run in polynomial time (see Lemmas 10.2, 10.6, 10.9, and 10.11), which also implies that each considered set $\mathcal{L}_{lead}$, $\mathcal{L}_{exp}$ and $\mathcal{L}_{cheap}$ has polynomial size.

Notice that all the solutions in $\mathcal{L}_{apx}$ (including APX) contain at most k centers, hence they are deterministically feasible. Consider one execution of the outer for loop. Suppose that the event from Lemma 10.6 holds for some $\Lambda \in \mathcal{L}_{lead}$, which happens with probability at least $1 - \varepsilon$, and focus on that choice of Λ in Line 5. Notice that under this condition Lemmas 10.7 and 10.8 hold. Suppose also that the event from Lemma 10.9 holds for some $\mathcal{Q} \in \mathcal{L}_{exp}$, which happens with probability at least $1 - 1/n$, and focus on that choice of $\mathcal{Q}$ in Line 7. Under these assumptions, which hold with probability at least $1 - 2\varepsilon$ altogether, there exists $\widetilde{\mathcal{U}} \in \mathcal{L}_{cheap}$ which satisfies the conditions of Lemma 10.11: let us focus on the execution of Line 9 for this value of $\widetilde{\mathcal{U}}$, and consider the corresponding solution $APX = S \cup \Lambda - \mathcal{Q} - \widetilde{\mathcal{U}}$. Observe that, under the above assumptions, Lemma 10.10 also holds. Thus

$$cost(APX) \le cost'(APX) \overset{Lem.10.11}{\le} cost'(S_{\mathcal{Q}\cup\mathcal{U}}) + O(\sqrt{\varepsilon})\text{opt} \overset{Lem.10.10}{\le} (2 + O(\sqrt{\varepsilon}))\text{opt}.$$

Since the outer for loop is executed $\frac{1}{2\varepsilon} \ln n$ many times, the probability that no $APX \in \mathcal{L}_{apx}$ is $2 + O(\sqrt{\varepsilon})$ approximate is at most $1/n$. The claim follows. □

Acknowledgement

The authors developed the new algorithm, found evidence that it might improve upon the previous work, and established the $(3 + \ln 2 + \varepsilon)$-approximation for the Euclidean case without any help of AI.

For the metric case, the authors developed a $(5+\varepsilon)$-analysis without any help of AI. The better $(4.9+\varepsilon)$-analysis was obtained by GPT 5.2/5.4 Pro/Thinking after several rounds of communication between the authors and it. The writeup for the $(4.9+\varepsilon)$-analysis has been verified and revised by the authors. The two analyses follow quite different routes within the spectral analysis framework and were found by different authors simultaneously.

Aditya Anand, Euiwoong Lee, and Amatya Sharma were supported in part by NSF grant 2236669. Ruiquan Gao was supported by NSF CCF-2112824 and Aviad Rubinstein's David and Lucile Packard Fellowship. Fabrizio Grandoni was partially supported by the SNF Grants 200021-200731 and 200021-236706. Ernest van Wijland was supported in part by the French PEPR integrated projects EPIQ (ANR-22-PETQ-0007).

References

[AGK+04] Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. "Local Search Heuristics for k-Median and Facility Location Problems". In: *SIAM J. Comput.* 33.3 (2004), pp. 544–562. DOI: 10.1137/S0097539702416402. URL: https://doi.org/10.1137/S0097539702416402.

[ANSW20] Sara Ahmadian, Ashkan Norouzi-Fard, Ola Svensson, and Justin Ward. "Better Guarantees for k-Means and Euclidean k-Median by Primal-Dual Algorithms". In: *SIAM J. Comput.* 49.4 (2020). DOI: 10.1137/18M1171321. URL: https://doi.org/10.1137/18M1171321.

[BF24] Niv Buchbinder and Moran Feldman. "Deterministic Algorithm and Faster Algorithm for Submodular Maximization subject to a Matroid Constraint". In: *CoRR* abs/2408.03583 (2024). DOI: 10.48550/ARXIV.2408.03583. arXiv: 2408.03583. URL: https://doi.org/10.48550/arXiv.2408.03583.

[BGH+26] Jaroslaw Byrka, Yuhao Guo, Yang Hu, Shi Li, Chengzhang Wan, and Zaixuan Wang. "k-Clustering via Iterative Randomized Rounding". In: *Proceedings of the 67th IEEE Symposium on Foundations of Computer Science, FOCS 2026*. To appear. 2026.

[BPR+15] Jaroslaw Byrka, Thomas W. Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. "An Improved Approximation for k-median, and Positive Correlation in Budgeted Optimization". In: *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*. SIAM, 2015, pp. 737–756. DOI: 10.1137/1.9781611973730.50. URL: https://doi.org/10.1137/1.9781611973730.50.

[CCG+25] Moses Charikar, Vincent Cohen-Addad, Ruiquan Gao, Fabrizio Grandoni, Euiwoong Lee, and Ernest van Wijland. "An Improved Greedy Approximation for (Metric) k-Means". In: *Proceedings of the 66th IEEE Symposium on Foundations of Computer Science, FOCS 2025*. 2025, pp. 233–240.

[CCG+26] Moses Charikar, Vincent Cohen-Addad, Ruiquan Gao, Fabrizio Grandoni, Euiwoong Lee, and Ernest van Wijland. "A $(4+\varepsilon)$-Approximation for Euclidean k-Means via Non-Monotone Dual-Fitting". In: *Proceedings of the 57th ACM Symposium on Theory of Computing, STOC 2026*. To appear. 2026.

[CCGG98] Moses Charikar, Chandra Chekuri, Ashish Goel, and Sudipto Guha. "Rounding via Trees: Deterministic Approximation Algorithms for Group Steiner Trees and k-Median". In: *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*. ACM, 1998, pp. 114–123. DOI: 10.1145/276698.276719. URL: https://doi.org/10.1145/276698.276719.

[CCPV11] Gruia Cualinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. "Maximizing a Monotone Submodular Function Subject to a Matroid Constraint". In: *SIAM J. Comput.* 40.6 (2011), pp. 1740–1766. DOI: 10.1137/080733991. URL: https://doi.org/10.1137/080733991.

[CEMN22] Vincent Cohen-Addad, Hossein Esfandiari, Vahab Mirrokni, and Shyam Narayanan. "Improved Approximations for Euclidean k-means and k-median, via Nested Quasi-Independent Sets". In: *Proceedings of the 54th ACM Symposium on Theory of Computing, STOC 2022*. 2022.

[CGL+25] Vincent Cohen-Addad, Fabrizio Grandoni, Euiwoong Lee, Chris Schwiegelshohn, and Ola Svensson. "A $(2+\varepsilon)$-Approximation Algorithm for Metric k-Median". In: *Proceedings of the 57th Annual ACM Symposium on Theory of Computing, STOC 2025, Prague, Czechia, June 23-27, 2025*. ACM, 2025, pp. 615–624. DOI: 10.1145/3717823.3718299. URL: https://doi.org/10.1145/3717823.3718299.

[CGLS23] Vincent Cohen-Addad Viallat, Fabrizio Grandoni, Euiwoong Lee, and Chris Schwiegelshohn. "Breaching the 2 LMP approximation barrier for facility location with applications to k-median". In: *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM. 2023, pp. 940–986.

[CGTS99] Moses Charikar, Sudipto Guha, Éva Tardos, and David B. Shmoys. "A Constant-Factor Approximation Algorithm for the k-Median Problem (Extended Abstract)". In: *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, May 1-4, 1999, Atlanta, Georgia, USA*. ACM, 1999, pp. 1–10. DOI: 10.1145/301250.301257. URL: https://doi.org/10.1145/301250.301257.

[CK19] Vincent Cohen-Addad and Karthik C. S. "Inapproximability of Clustering in Lp Metrics". In: *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*. IEEE Computer Society, 2019, pp. 519–539. DOI: 10.1109/FOCS.2019.00040. URL: https://doi.org/10.1109/FOCS.2019.00040.

[CKM16] Vincent Cohen-Addad, Philip N. Klein, and Claire Mathieu. "Local Search Yields Approximation Schemes for k-Means and k-Median in Euclidean and Minor-Free Metrics". In: *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, Hyatt Regency, New Brunswick, New Jersey, USA, October 9-11, 2016*. IEEE Computer Society, 2016, pp. 353–364. DOI: 10.1109/FOCS.2016.46. URL: https://doi.org/10.1109/FOCS.2016.46.

[CL12] Moses Charikar and Shi Li. "A Dependent LP-Rounding Approach for the k-Median Problem". In: *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part I*. Vol. 7391. Springer, 2012, pp. 194–205.

[Coh18] Vincent Cohen-Addad. "A Fast Approximation Scheme for Low-Dimensional k-Means". In: *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*. SIAM, 2018, pp. 430–440. DOI: 10.1137/1.9781611975031.29. URL: https://doi.org/10.1137/1.9781611975031.29.

[CSL22] Vincent Cohen-Addad, Karthik C S, and Euiwoong Lee. "Johnson coverage hypothesis: Inapproximability of k-means and k-median in lp-metrics". In: *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM. 2022, pp. 1493–1530.

[DKKR03] W Fernandez De La Vega, Marek Karpinski, Claire Kenyon, and Yuval Rabani. "Approximation schemes for clustering problems". In: *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*. 2003, pp. 50–58.

[FRS16] Zachary Friggstad, Mohsen Rezapour, and Mohammad R. Salavatipour. "Local Search Yields a PTAS for k-Means in Doubling Metrics". In: *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, Hyatt Regency, New Brunswick, New Jersey, USA, October 9-11, 2016*. IEEE Computer Society, 2016, pp. 365–374. DOI: 10.1109/FOCS.2016.47. URL: https://doi.org/10.1109/FOCS.2016.47.

[FW14] Yuval Filmus and Justin Ward. "Monotone Submodular Maximization over a Matroid via Non-Oblivious Local Search". In: *SIAM J. Comput.* 43.2 (2014), pp. 514–542. DOI: 10.1137/130920277. URL: https://doi.org/10.1137/130920277.

[GK99] Sudipto Guha and Samir Khuller. "Greedy strikes back: Improved facility location algorithms". In: *Journal of algorithms* 31.1 (1999), pp. 228–248.

[GOR+22] Fabrizio Grandoni, Rafail Ostrovsky, Yuval Rabani, Leonard J. Schulman, and Rakesh Venkat. "A refined approximation for Euclidean k-means". In: *Inf. Process. Lett.* 176 (2022), p. 106251. DOI: 10.1016/J.IPL.2022.106251. URL: https://doi.org/10.1016/j.ipl.2022.106251.

[GPST23] Kishen N. Gowda, Thomas W. Pensyl, Aravind Srinivasan, and Khoa Trinh. "Improved Bi-point Rounding Algorithms and a Golden Barrier for k-Median". In: *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*. SIAM, 2023, pp. 987–1011. DOI: 10.1137/1.9781611977554.CH38. URL: https://doi.org/10.1137/1.9781611977554.ch38.

[GT08] Anupam Gupta and Kanat Tangwongsan. "Simpler Analyses of Local Search Algorithms for Facility Location". In: *CoRR* abs/0809.2554 (2008). arXiv: 0809.2554. URL: http://arxiv.org/abs/0809.2554.

[JMM+03] Kamal Jain, Mohammad Mahdian, Evangelos Markakis, Amin Saberi, and Vijay V. Vazirani. "Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP". In: *J. ACM* 50.6 (2003), pp. 795–824. DOI: 10.1145/950620.950621. URL: https://doi.org/10.1145/950620.950621.

[JMS02] Kamal Jain, Mohammad Mahdian, and Amin Saberi. "A new greedy approach for facility location problems". In: *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*. ACM, 2002, pp. 731–740. DOI: 10.1145/509907.510012. URL: https://doi.org/10.1145/509907.510012.

[JV01] K. Jain and V. Vazirani. "Approximation algorithms for metric facility location and k-Median problems using the primal-dual schema and Lagrangian relaxation". In: *J. ACM* 48.2 (2001), pp. 274–296. DOI: 10.1145/375827.375845. URL: http://doi.acm.org/10.1145/375827.375845.

[KMN+04] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. "A local search approximation algorithm for k-means clustering". In: *Comput. Geom.* 28.2-3 (2004), pp. 89–112. DOI: 10.1016/j.comgeo.2004.03.003. URL: https://doi.org/10.1016/j.comgeo.2004.03.003.

[LS13] Shi Li and Ola Svensson. "Approximating k-median via pseudo-approximation". In: *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*. 2013, pp. 901–910. DOI: 10.1145/2488608.2488723. URL: http://doi.acm.org/10.1145/2488608.2488723.

[LS25] Euiwoong Lee and Kijun Shin. "Facility Location on High-Dimensional Euclidean Spaces". In: *16th Innovations in Theoretical Computer Science Conference (ITCS 2025)*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. 2025, pp. 70–1.

[Mat00] Jií Matouek. "On approximate geometric k-clustering". In: *Discrete & Computational Geometry* 24.1 (2000), pp. 61–84.

[Mey00] C.D. Meyer. *Matrix Analysis and Applied Linear Algebra*. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 2000. ISBN: 9780898719512. URL: https://books.google.com/books?id=-7JeAwAAQBAJ.

A Proof of Theorems 1.1 and 1.2

Theorem 1.1. *For any $\varepsilon > 0$, there is a polynomial-time algorithm that yields a $(3 + \ln 2 + \varepsilon)$-approximation for Euclidean k-Means.*

Theorem 1.2. *For any $\varepsilon > 0$, there is a polynomial-time algorithm that yields a $(4.9 + \varepsilon)$-approximation for Metric k-Means.*

Proofs of Theorems 1.1 and 1.2. We compute a set of feasible solutions, and return the cheapest one. Let Δ be the maximum extra number of centers computed by the algorithm from Theorem 1.3 (this number is independent from k). Let $k' = \max\{1, k - \Delta\}$. Obtain a solution by computing the optimum solution with one center if $k' = 1$, and otherwise by running the algorithm from Theorem 1.3 with target number of centers being k' (notice that it returns a solution with at most k centers, and is hence feasible).

We run the algorithm from either of Theorems 1.4 and 1.5 for every integer $k'' \in (k', k]$ (thus obtaining solutions with $k'' \leq k$ centers, hence feasible).

If $\mathrm{OPT}_{k'} \leq (1 + \varepsilon)\mathrm{OPT}_k$, the first solution is $(1 + \varepsilon)\Gamma$ approximate. Otherwise, $\mathrm{OPT}_{k'} > (1 + \varepsilon)\mathrm{OPT}_k$. Thus there exists an integer $k'' \in (k', k]$ such that $\mathrm{OPT}_{k''} \leq (1 + \varepsilon)\mathrm{OPT}_k$

and $\mathrm{OPT}_{k''-1} \geq (1+\beta)\mathrm{OPT}_{k''}$ for $\beta = \Omega(\varepsilon/\Delta) = \Omega(\varepsilon^4/\log n)$. For that value of k'' the corresponding instance of Metric k-Means is β-stable, hence we can compute a solution with cost at most $(4+O(\sqrt{\varepsilon}))\mathrm{OPT}_{k''} \leq (1+\varepsilon)(4+O(\sqrt{\varepsilon}))\mathrm{OPT}_k$ using Theorem 1.4, which yields a $(1+\varepsilon)(4+\sqrt{\varepsilon})$ approximation. Similarly, in the Euclidean case, we get a $(1+\varepsilon)(2+O(\sqrt{\varepsilon}))$ approximation using Theorem 1.5. The proof is then complete after suitably rescaling ε. □

B Missing Proofs in Section 5

B.1 Proof of Lemma 5.8

In this appendix, we prove Lemma 5.8 for the sake of completeness.

Lemma 5.8 (essentially, [CCG+26]). *For any $w : [s] \to \mathbb{R}_{>0}$ and any $x_1, \ldots, x_s \in \mathbb{R}$, we have*

$$\sum_{j<\ell} w(j)P_{j\ell}(x_j - x_\ell)^2 \leq \sum_{j\in[s]} w(j)x_j^2 \ .$$

B.1.1 Preliminaries on weighted k-means

We define the weighted versions of Euclidean k-Means that will be useful later in the proof.

Definition B.1 (weighted discrete Euclidean k-Means). *The input consists of a set of clients D, a set of facilities $\mathcal{F}$ and their locations $x : D \cup \mathcal{F} \to \mathbb{R}^{\mathsf{dim}}$. In addition there is a weight function $w : D \to \mathbb{R}_{>0}$ mapping each client to a positive weight. The goal of the discrete Euclidean k-Means problem is to find a set of k centers $S \subseteq \mathcal{F}$ such that $|S| = k$ and the following objective is minimized:*

$$\sum_{j\in D} w(j) \min_{i\in S} \|x_i - x_j\|_2^2 \ .$$

Definition B.2 (weighted continuous Euclidean k-Means). *The input consists of a set of clients D and their locations $x : D \to \mathbb{R}^{\mathsf{dim}}$. In addition there is a weight function $w : D \to \mathbb{R}_{>0}$ mapping each client to a positive weight. The goal of the weighted continuous Euclidean k-Means problem is to find a set of k centers $S \subseteq \mathbb{R}^{\mathsf{dim}}$ such that $|S| = k$ and the following objective is minimized:*

$$\sum_{j\in D} w(j) \cdot \min_{c\in S} \|c - x_j\|_2^2 \ .$$

In particular, an instance is unweighted if $w(j) = 1$ for any client $j \in D$.

One technical lemma we will use later is a characterization of the optimal cost of a cluster consisting of a set of clients among all possible choices of the center.

Lemma B.3 (single-cluster cost of weighted continuous Euclidean k-Means). *Suppose the set $C \subseteq D$ forms a cluster. Suppose that $W = \sum_{j\in C} w(j)$. The optimal center is the weighted average of the points $\sum_{j\in C} \frac{w(j)}{W} x_j$, and the cost is*

$$\mathsf{cost}_w(C) = \frac{1}{W} \cdot \sum_{j,\ell\in\binom{C}{2}} w(j)w(l)\|x_j - x_\ell\|_2^2$$

In particular, if the instance is unweighted, we will use the notation cost(C) for simplicity.

Proof. The optimal center is given by optimizing the following the total weighted cost

$$f(c) = \sum_{j \in C} w(j) \cdot \|c - x_j\|_2^2 = \sum_{j \in C} w(j) \cdot (\|c\|_2^2 - 2\langle x_j, c\rangle + \|x_j\|_2^2)$$
$$= W\|c\|_2^2 - 2 \cdot \sum_{j \in C} \langle w(j) \cdot x_j, c\rangle + \sum_{j \in C} w(j) \cdot \|x_j\|_2^2 \ .$$

It is clear that $f(c)$ is convex and the gradient is

$$\nabla f(c) = 2Wc - 2 \cdot \sum_{j \in C} w(j) \cdot x_j \ .$$

Therefore, the optimal center is $c^* = \frac{\sum_{j \in C} w(j) \cdot x_j}{W}$. In particular, the total weighted cost:

$$f(c^*) = -\frac{1}{W} \cdot \left\| \sum_{j \in C} w(j) \cdot x_j \right\|_2^2 + \sum_{j \in C} w(j) \cdot \|x_j\|_2^2$$
$$= \frac{1}{W} \cdot \sum_{j,\ell \in C} w(j) \cdot w(\ell) \cdot (\|x_j\|_2^2 - \langle x_j, x_\ell\rangle)$$
$$= \frac{1}{W} \cdot \sum_{j,\ell \in \binom{C}{2}} w(j) \cdot w(\ell) \cdot (\|x_j\|_2^2 - 2\langle x_j, x_\ell\rangle + \|x_\ell\|_2^2)$$
$$= \frac{1}{W} \cdot \sum_{j,\ell \in \binom{C}{2}} w(j) \cdot w(\ell) \cdot \|x_j - x_\ell\|_2^2 \ .$$

This completes the proof. □

Lemma B.4. *For any sets $C_1 \subseteq C_2$ of clients and any weight function w, we have $cost_w(C_1) \leq cost_w(C_2)$.*

Proof. Suppose c_2 is the optimal center for C_2. Then, we have

$$\mathrm{cost}_w(C_1) \leq \sum_{j \in C_1} w(j) \cdot \|x_j - c_2\|_2^2 \leq \sum_{j \in C_2} w(j) \cdot \|x_j - c_2\|_2^2 = \mathrm{cost}_w(C_2) \ .$$ □

Lemma B.5. *Consider any set C of clients, any client $j \notin C$ and any weight function w. Let $W = \sum_{\ell \in C} w(\ell)$. Then, we have*

$$cost_w(C \cup \{j\}) = \frac{W}{W + w(j)} cost_w(C) + \frac{w(j)}{W + w(j)} \cdot \sum_{\ell \in C} w(\ell)\|x_j - x_\ell\|_2^2 \ .$$

Proof. Using Lemma B.3, we have

$$\mathrm{cost}_w(C \cup \{j\}) = \frac{1}{W + w(j)} \cdot \sum_{\ell,\ell' \in \binom{C}{2}} w(\ell)w(\ell')\|x_\ell - x_{\ell'}\|_2^2 + \frac{w(j)}{W + w(j)} \cdot \sum_{\ell \in C} w(\ell)\|x_j - x_\ell\|_2^2$$
$$= \frac{W}{W + w(j)} \mathrm{cost}_w(C) + \frac{w(j)}{W + w(j)} \cdot \sum_{\ell \in C} w(\ell)\|x_j - x_\ell\|_2^2 \ .$$ □

B.1.2 The proof

Let $a \in [s-1]$ be the smallest integer such that $a \in DC^t$ for some $t \in [s]$. If no such a exists, we set $a = s$ (note that otherwise $a < s$). We will prove by induction on a.

Inductive hypothesis. For any $s' \in [s]$ and any client $j \in [s]$, we let $w_{s'}(j) = \sum_{t\in[s']} \beta_t \eta_{tj}$. The inductive hypothesis is then: for any $s' \in [s]$, we have

$$\sum_{t\in[s']} \beta_t \sum_{j\in DC^t} w(j) \sum_{\ell\in IC^t\cup\{t\}} p_{tj\ell} \cdot \|x_j - x_\ell\|^2 \le \mathrm{cost}_{w_{s'}}([a:s]) \ . \tag{49}$$

Because of recurrence relation of β_t (Equation (15)), we have $w_s(j) = w(j)$ for each $j \in [s]$. We also have $P_{j\ell} = \sum_{t\in[s]:j\in DC^t} \beta_t p_{tj\ell}$. Therefore, the LHS of Equation (49) equals $\sum_{j\in[s]} w(j) \sum_{\ell\in[s]} P_{j\ell} \cdot \|x_j - x_\ell\|^2$. Lemma 5.8 then follows from the inductive hypothesis for $s' = s$:

$$\sum_{j<\ell} P_{j\ell} \cdot \|x_j - x_\ell\|^2 \le \mathrm{cost}_w([a:s]) \le \mathrm{cost}_w([s]) \le \sum_{j\in[s]} w(j)\|x_j\|^2 \ ,$$

where the last inequality follows from the fact that $\sum_{j\in[s]} w(j)\|x_j\|^2$ is the total weighted cost of opening the center at 0^{dim}.

Base case. The base case is when $a = s$. In this case, because $DC^t = \emptyset$ for any $t \in [s]$, we have $p_{tj\ell} = 0$ for any $t, j, \ell \in [s]$, $j \in DC^t$. The proof of the base case is then straightforward because the LHS of Equation (49) equals 0.

Inductive step. We assume that the inductive hypothesis holds for any $a' > a$ and we will prove that it also holds for a. Let t_a the smallest index t such that $a \in DC^t$. Note that we have $a < t_a \le s$ because $DC^a \subseteq [a-1]$ and $a \in DC^t$ for some $t \in [s]$.

Consider the case $(\widetilde{DC}, \widetilde{IC}, w)$, where the weights are the same, but for each $t \in [s]$, we have $\widetilde{DC}^t = DC^t \setminus \{a\}$ and $\widetilde{IC}^t = IC^t \cup \{a\}$. We will use $\tilde{p}_{tj\ell}, \tilde{\eta}_{tj}, \tilde{\beta}_t, \tilde{a}$ to denote the corresponding $p_{tj\ell}, \eta_{tj}, \beta_t, a$ in the case $(\widetilde{DC}, \widetilde{IC}, w)$. In this case, we have $\tilde{a} > a$. Let $\tilde{w}_{s'}(j) = \sum_{t\in[s']} \tilde{\beta}_t \tilde{\eta}_{tj}$ for any $s' \in [s]$ and any client $j \in [s]$. Hence, according to the inductive hypothesis, we have

$$\sum_{t\in[s']} \tilde{\beta}_t \sum_{j\in\widetilde{DC}^t} w(j) \sum_{\ell\in\widetilde{IC}^t\cup\{t\}} \tilde{p}_{tj\ell} \cdot \|x_j - x_\ell\|^2 \le \mathrm{cost}_{\tilde{w}_{s'}}([\tilde{a}:s])$$

Next, we compare the parameters in the case (DC, IC, w) and the case $(\widetilde{DC}, \widetilde{IC}, w)$.

- For $t < t_a$, we have $\widetilde{DC}^t = DC^t$ and $\widetilde{IC}^t = IC^t$. Therefore, we have $\tilde{p}_{tj\ell} = p_{tj\ell}$ for any $j, \ell \in [s]$.
- For $t \ge t_a$, we have $DC^t = \widetilde{DC}^t \cup \{a\}$ and $\widetilde{IC}^t = IC^t \setminus \{a\}$. Therefore, we have the following properties for such t.
  - We have $\pi_t(j) = \frac{\mathsf{suf}(a)}{\mathsf{suf}(a+1)} \cdot \tilde{\pi}_t(j)$ for any $j \in [a+1:s]$ and $\pi_t(j) = \tilde{\pi}_t(j) = 1$ for $j \in [a]$.
  - We have $\eta_{tj} = \frac{\mathsf{suf}(a)}{\mathsf{suf}(a+1)} \cdot \tilde{\eta}_{tj}$ for any $j \in [a+1:s]$. We have $\eta_{ta} = 0$, $\tilde{\eta}_{ta} = 1$. We have $\eta_{tj} = \tilde{\eta}_{tj} = 1$ for any $j \in [a-1]$.

– We have $\sigma_t(j) = \frac{\mathsf{suf}(a)}{\mathsf{suf}(a+1)} \cdot \tilde{\sigma}_t(j)$ for any $j \in [a+1:s]$ and $\sigma_t(j) = \tilde{\sigma}_t(j)$ for $j \in [a]$. In particular, we have $\sigma_t(a) = \mathsf{suf}(a)$.

– We have $p_{tj\ell} = \tilde{p}_{tj\ell}$ for any $j \in [a+1:s] \cap DC^t$ and $\ell \in IC^t \cup \{t\}$. We have $p_{ta\ell} = \eta_{t\ell}/\mathsf{suf}(a) = \tilde{\eta}_{t\ell}/\mathsf{suf}(a+1)$ and $\tilde{p}_{ta\ell} = 0$ for any $\ell \in [a+1:s] \cap IC^t$. We have $p_{tj\ell} = \tilde{p}_{tj\ell} = 0$ for all other cases.

Using the above properties, we have $\beta_t = \frac{\mathsf{suf}(a+1)}{\mathsf{suf}(a)} \cdot \tilde{\beta}_t$ for $t \geq t_a$ and $\beta_t = \tilde{\beta}_t$ for $a < t < t_a$. In addition, we have $\beta_a = 1 - \sum_{j>a} \beta_j$. It can be verified by the recurrence relation of β_t (Equation (15)). Therefore, we have $\beta_t \eta_{tj} = \tilde{\beta}_t \tilde{\eta}_{tj}$ for any $t \in [s]$, $j \in [a+1:s]$. This implies that $w_{s'}(j) = \tilde{w}_{s'}(j)$ for any $s' \in [s]$ and any $j \in [a+1:s]$.

case 1: $s' < t_a$**.** In this case, we have $DC^t = \widetilde{DC}^t$ for any $t \in [s']$, and we have $\beta_t = \tilde{\beta}_t$ for any $t \in [s']$ except for $t = a$ (if $s' \geq a$). Because $DC^a = \emptyset$ and Lemma B.4, we have

$$\sum_{t\in[s']} \beta_t \sum_{j\in DC^t} w(j) \sum_{\ell\in IC^t\cup\{t\}} p_{tj\ell}\|x_j - x_\ell\|^2 = \sum_{t\in[s']} \tilde{\beta}_t \sum_{j\in\widetilde{DC}^t} w(j) \sum_{\ell\in\widetilde{IC}^t\cup\{t\}} \tilde{p}_{tj\ell}\|x_j - x_\ell\|^2$$
$$\leq \mathrm{cost}_{\tilde{w}_{s'}}([\tilde{a}:s]) = \mathrm{cost}_{w_{s'}}([\tilde{a}:s]) \leq \mathrm{cost}_{w_{s'}}([a:s]).$$

case 2: $s' \geq t_a$**.** In this case, we have $w_{s'}(a) = w(a)$ because $\eta_{ta} = 0$ for any $t > s'$. Using the above properties, we have

$$\begin{aligned}
&\sum_{t\in[s']} \beta_t \sum_{j\in DC^t} w(j) \sum_{\ell\in IC^t\cup\{t\}} p_{tj\ell}\|x_j - x_\ell\|^2 \\
=& \sum_{t\in[t_a-1]} \beta_t \sum_{j\in DC^t} w(j) \sum_{\ell\in IC^t\cup\{t\}} p_{tj\ell}\|x_j - x_\ell\|^2 + \sum_{t\in[t_a:s']} \beta_t \sum_{j\in DC^t} w(j) \sum_{\ell\in IC^t\cup\{t\}} p_{tj\ell}\|x_j - x_\ell\|^2 \\
=& \sum_{t\in[t_a-1]} \tilde{\beta}_t \sum_{j\in\widetilde{DC}^t} w(j) \sum_{\ell\in\widetilde{IC}^t\cup\{t\}} \tilde{p}_{tj\ell}\|x_j - x_\ell\|^2 + \sum_{t\in[t_a:s']} \beta_t w(a) \sum_{\ell\in IC^t\cup\{t\}} p_{ta\ell}\|x_a - x_\ell\|^2 \\
&+ \sum_{t\in[t_a:s']} \frac{\mathsf{suf}(a+1)}{\mathsf{suf}(a)} \tilde{\beta}_t \sum_{j\in\widetilde{DC}^t} w(j) \sum_{\ell\in\widetilde{IC}^t\cup\{t\}} \tilde{p}_{tj\ell}\|x_j - x_\ell\|^2 && (\tilde{p}_{tja} = 0) \\
=& \frac{w(a)}{\mathsf{suf}(a)} \sum_{t\in[t_a-1]} \tilde{\beta}_t \sum_{j\in\widetilde{DC}^t} w(j) \sum_{\ell\in\widetilde{IC}^t\cup\{t\}} \tilde{p}_{tj\ell}\|x_j - x_\ell\|^2 + \sum_{t\in[t_a:s']} \beta_t w(a) \sum_{\ell\in IC^t\cup\{t\}} p_{ta\ell}\|x_a - x_\ell\|^2 \\
&\frac{\mathsf{suf}(a+1)}{\mathsf{suf}(a)} \sum_{t\in[s']} \tilde{\beta}_t \sum_{j\in\widetilde{DC}^t} w(j) \sum_{\ell\in\widetilde{IC}^t\cup\{t\}} \tilde{p}_{tj\ell}\|x_j - x_\ell\|^2 \\
\leq& \frac{w(a)}{\mathsf{suf}(a)} \cdot \mathrm{cost}_{w_{t_a-1}}([\tilde{a}:s]) + \frac{\mathsf{suf}(a+1)}{\mathsf{suf}(a)} \cdot \mathrm{cost}_{w_{s'}}([\tilde{a}:s]) + \underbrace{\sum_{t\in[t_a:s']} \beta_t w(a) \sum_{\ell\in IC^t\cup\{t\}} p_{ta\ell}\|x_a - x_\ell\|^2}_{S_3} \\
&\qquad\qquad \text{(inductive hypothesis for } \tilde{a} \text{ and } w_\cdot(j) = \tilde{w}_\cdot(j) \text{ for } j \in [a+1:s])
\end{aligned}$$

Next, we consider the new term S_3. We have

$$\begin{aligned}
S_3 &= \sum_{t\in[t_a:s']} \beta_t w(a) \sum_{\ell\in IC^t\cup\{t\}} p_{ta\ell}\|x_a - x_\ell\|^2 \\
&= \frac{\mathsf{suf}(a+1)w(a)}{\mathsf{suf}(a)} \sum_{t\in[t_a:s']} \tilde{\beta}_t \sum_{\ell>a:\ell\in IC^t\cup\{t\}} \frac{\tilde{\eta}_{t\ell}}{\mathsf{suf}(a+1)}\|x_a - x_\ell\|^2 \ . \\
&= \frac{w(a)}{\mathsf{suf}(a)} \sum_{t\in[t_a:s']} \tilde{\beta}_t \sum_{\ell>a} \tilde{\eta}_{t\ell}\|x_a - x_\ell\|^2 \qquad (\tilde{\eta}_{t\ell} = 0 \text{ for } \ell \in DC^t) \\
&= \frac{w(a)}{\mathsf{suf}(a)} \sum_{\ell>a} \|x_a - x_\ell\|^2 \cdot \sum_{t\in[t_a:s']} \tilde{\beta}_t\tilde{\eta}_{t\ell} \\
&= \frac{w(a)}{\mathsf{suf}(a)} \sum_{\ell>a} \|x_a - x_\ell\|^2 \cdot \left(\sum_{t\in[s']} \tilde{\beta}_t\tilde{\eta}_{t\ell} - \sum_{t\in[t_a-1]} \tilde{\beta}_t\tilde{\eta}_{t\ell}\right) \\
&= \frac{w(a)}{\mathsf{suf}(a)} \sum_{\ell>a} \|x_a - x_\ell\|^2 \cdot (\tilde{w}_{s'}(\ell) - \tilde{w}_{t_a-1}(\ell)) \\
&= \frac{w_{s'}(a)}{\mathsf{suf}(a)} \sum_{\ell>a} w_{s'}(\ell)\|x_a - x_\ell\|^2 - \frac{w(a)}{\mathsf{suf}(a)} \sum_{\ell>a} w_{t_a-1}(\ell)\|x_a - x_\ell\|^2 \ .
\end{aligned}$$

According to Lemmas B.4 and B.5, we have

$$\begin{aligned}
&\frac{w_{s'}(a)}{\mathsf{suf}(a)} \sum_{\ell>a} w_{s'}(\ell)\|x_a - x_\ell\|^2 + \frac{\mathsf{suf}(a+1)}{\mathsf{suf}(a)} \cdot \mathrm{cost}_{w_{s'}}([\tilde{a} : s]) \\
&\quad\le \frac{w_{s'}(a)}{\mathsf{suf}(a)} \sum_{\ell>a} w_{s'}(\ell)\|x_a - x_\ell\|^2 + \frac{\mathsf{suf}(a+1)}{\mathsf{suf}(a)} \cdot \mathrm{cost}_{w_{s'}}([a+1 : s]) = \mathrm{cost}_{w_{s'}}([a : s]) \ .
\end{aligned}$$

Note that $\sum_{t>a} w_{t_a-1}(t)\|x_a - x_t\|^2$ is the cost of the cluster $[a+1 : s]$ with the center at x_a. Therefore, we have $\sum_{t>a} w_{t_a-1}(t)\|x_a - x_t\|^2 \ge \mathrm{cost}_{w_{t_a-1}}([a+1 : s])$. Therefore, putting S_3 back to the earlier inequality, we have completed the inductive step:

$$\sum_{t\in[s']} \beta_t \sum_{j\in DC^t} w(j) \sum_{\ell\in IC^t\cup\{t\}} p_{tj\ell}\|x_j - x_\ell\|^2 \le \mathrm{cost}_{w'_s}([a : s]) \ .$$

B.2 Proof of Lemma 5.20

In the rest of this subsection, we will prove the above lemma. We will use the same set of non-negative parameters $p_{tj\ell}$ and β_t as in the proof of Lemma 4.11 (in terms of $(w(t))_{t\in[s]}$, $(DC^t)_{t\in[s]}$ and $(IC^t)_{t\in[s]}$). One property we will use about the β-parameters is the following:

Lemma B.6. *For any $t \in [a+1 : s-1]$, we have $\beta_t = 0$.*

Proof. For any $t \in [a+1 : s-1]$, we have $t \in IC^s$. Therefore, according to our definition of $\pi_t(\ell)$, we have $\pi_t(\ell) = \pi_t(s)$ for any $\ell \in [a+1 : s]$. Further, $\eta_{s\ell} = w(\ell)\pi_t(s)$ for any $\ell \in [a+1 : s]$. Note that β_s is defined by $\beta_s\eta_{ss} = w(s)$. We have $\beta_s = 1/\pi_t(s)$. Therefore, for any $\ell \in [a+1 : s-1]$, we have $\beta_s\eta_{s\ell} = w(\ell)$. Using the recurrence relation in Equation (15), we have $\beta_t\eta_{t\ell} = 0$ for any $\ell \in [a+1 : s-1]$. Because $\eta_{tt} > 0$, we have $\beta_t = 0$ for any $t \in [a+1 : s-1]$. □

We also consider adding one term that is no greater than 0 to the LHS of the third constraint (3) for each $t \in [a] \cup \{s\}$, using the same parameters $p_{tj\ell}$ as in the proof of Lemma 4.11. More specifically, we will consider the term on the LHS of Equation (50) in the following lemma. This is only slightly different from the term we considered in the proof of Lemma 4.11, with a $(1+2\varepsilon)$ factor before each $(r_{jt}+d(j,\ell))^2$.

Lemma B.7. *For each $t \in [a] \cup \{s\}$, we have*

$$\sum_{j \in DC^t} w(j) \cdot \sum_{\ell \in IC^t \cup \{t\}} p_{tj\ell}(\alpha^*_\ell - (1+2\varepsilon)(r_{jt}+d(j,\ell))^2) \le 0 \ . \tag{50}$$

Proof. Because the second constraint (2) holds for $t \in [a]$, it is straightforward to verify that the above term is upper bounded by 0 for $t \in [a]$. Next, we will consider the case when $t = s$.

Let $W^I = \sum_{\ell \in [a+1:s]} w(\ell)$ and $W^L = \sum_{\ell \in L} w(\ell)$. According to the first bullet of the fourth constraint (4), we have $W^L \le \varepsilon \cdot W^I$. Using the second and the third bullet of the fourth constraint, we have

$$\begin{aligned} 0 \ge & \sum_{j \in DC^s} w(j) \cdot \sum_{\ell \in (IC^s \cup \{s\}) \setminus L} p_{sj\ell}(\alpha^*_\ell - (1+\varepsilon)(r_{js}+d(j,\ell))^2) \\ & + \sum_{j \in DC^s} w(j) \sum_{\ell \in L} p_{sj\ell} \left(\alpha^*_\ell - \sum_{\ell' \in [a+1:s]} \frac{w(\ell')}{W^I} \cdot (r_{js}+d(j,\ell'))^2 \right) \end{aligned} \tag{51}$$

For the second part of the RHS of the above inequality, we have

$$\begin{aligned} & \sum_{j \in DC^s} w(j) \sum_{\ell \in L} p_{sj\ell} \left(\alpha^*_\ell - \sum_{\ell' \in [a+1:s]} \frac{w(\ell')}{W^I} \cdot (r_{js}+d(j,\ell'))^2 \right) \\ = & \sum_{j \in DC^s} w(j) \sum_{\ell \in L} p_{sj\ell}\alpha^*_\ell - \sum_{j \in DC^s} w(j) \sum_{\ell \in L} p_{sj\ell} \sum_{\ell' \in [a+1:s]} \frac{w(\ell')}{W^I} \cdot (r_{js}+d(j,\ell'))^2 \\ = & \sum_{j \in DC^s} w(j) \sum_{\ell \in L} p_{sj\ell}\alpha^*_\ell - \sum_{j \in DC^s} w(j) \sum_{\ell' \in [a+1:s]} \left(\sum_{\ell \in L} p_{sj\ell} \right) \cdot \frac{w(\ell')}{W^I} \cdot (r_{js}+d(j,\ell'))^2 \ . \end{aligned}$$

Note that if $\ell \in [a], \ell' \in [a+1:s]$, we have $p_{sj\ell} \le \frac{w(\ell)}{w(\ell')} \cdot p_{sj\ell'}$ according to our definition of $p_{tj\ell}$ (see Equations (11) and (12)). Therefore, we have

$$\begin{aligned} & \sum_{j \in DC^s} w(j) \sum_{\ell \in L} p_{sj\ell} \left(\alpha^*_\ell - \sum_{\ell' \in [a+1:s]} \frac{w(\ell')}{W^I} \cdot (r_{js}+d(j,\ell'))^2 \right) \\ \ge & \sum_{j \in DC^s} w(j) \sum_{\ell \in L} p_{sj\ell}\alpha^*_\ell - \sum_{j \in DC^s} w(j) \sum_{\ell' \in [a+1:s]} p_{sj\ell'} \left(\sum_{\ell \in L} \frac{w(\ell)}{w(\ell')} \right) \cdot \frac{w(\ell')}{W^I} \cdot (r_{js}+d(j,\ell'))^2 \\ = & \sum_{j \in DC^s} w(j) \sum_{\ell \in L} p_{sj\ell}\alpha^*_\ell - \sum_{j \in DC^s} w(j) \sum_{\ell' \in [a+1:s]} p_{sj\ell'} \cdot \frac{W^L}{W^I} \cdot (r_{js}+d(j,\ell'))^2 \\ \ge & \sum_{j \in DC^s} w(j) \sum_{\ell \in L} p_{sj\ell}\alpha^*_\ell - \sum_{j \in DC^s} w(j) \sum_{\ell' \in [a+1:s]} \varepsilon \cdot p_{sj\ell'} (r_{js}+d(j,\ell'))^2 \end{aligned}$$

Putting this back to Equation (51), we have

$$\begin{aligned}
0 &\geq \sum_{j\in DC^s} w(j)\cdot \sum_{\ell\in(IC^s\cup\{s\})\setminus L} p_{sj\ell}(\alpha^*_\ell - (1+\varepsilon)(r_{js}+d(j,\ell))^2) \\
&\qquad + \sum_{j\in DC^s} w(j)\sum_{\ell\in L} p_{sj\ell}\alpha^*_\ell - \sum_{j\in DC^s} w(j) \sum_{\ell'\in[a+1:s]} \varepsilon\cdot p_{sj\ell'}(r_{js}+d(j,\ell'))^2 \\
&\geq \sum_{j\in DC^s} w(j)\cdot \sum_{\ell\in(IC^s\cup\{s\})} p_{sj\ell}(\alpha^*_\ell - (1+2\varepsilon)(r_{js}+d(j,\ell))^2)
\end{aligned}$$

Hence, we have completed the proof of this lemma. □

By adding the LHS of Equation (50) to the LHS of the third constraint (3) for each $t\in[a]\cup\{s\}$, we have the following inequality (for each $t\in[a]\cup\{s\}$):

LHS of constraint (3)

$$\begin{aligned}
&\geq \sum_{\ell\in[t:s]} w(\ell)\alpha^*_t + \sum_{\ell\in IC^t} w(\ell)\alpha^*_\ell + \sum_{j\in DC^t} w(j)\left(r^2_{jt} + \sum_{\ell\in IC^t\cup\{t\}} p_{tj\ell}(\alpha^*_\ell - (1+2\varepsilon)(r_{jt}+d(j,\ell))^2)\right) \\
&= \sum_{\ell\in[t:s]} w(\ell)\alpha^*_t + \sum_{\ell\in IC^t} w(\ell)\alpha^*_\ell + \sum_{j\in DC^t} w(j) \sum_{\ell\in IC^t\cup\{t\}} p_{tj\ell}\alpha^*_\ell \\
&\qquad - \sum_{j\in DC^t} w(j)\left(2\varepsilon r^2_{jt} + \sum_{\ell\in IC^t\cup\{t\}} p_{tj\ell}(1+2\varepsilon)(2r_{jt}d(j,\ell)+d^2(j,\ell))\right) \qquad (\textstyle\sum_\ell p_{tj\ell}=1) \\
&= \sum_{\ell\in IC^t\cup\{t\}} \eta_{t\ell}\alpha^*_\ell - \sum_{j\in DC^t} w(j)\left(2\varepsilon r^2_{jt} + \sum_{\ell\in IC^t\cup\{t\}} p_{tj\ell}(1+2\varepsilon)(2r_{jt}d(j,\ell)+d^2(j,\ell))\right) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(Corollary 5.5)} \\
&\geq \sum_{\ell\in IC^t\cup\{t\}} \eta_{t\ell}\alpha^*_\ell - \sum_{j\in DC^t} w(j)\left(2\varepsilon R^2_j + \sum_{\ell\in IC^t\cup\{t\}} p_{tj\ell}(1+2\varepsilon)(2R_jd(j,\ell)+d^2(j,\ell))\right) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (r_{jt}\leq R_j)
\end{aligned}$$

Despite not having the above inequality of $t\in[a+1:s-1]$, because we have $\beta_t=0$ for any $t\in[a+1:s-1]$ (Lemma B.6), we have the following inequality for any $t\in[s]$:

$$\begin{aligned}
&\beta_t\cdot\left[\sum_{\ell\in IC^t\cup\{t\}} \eta_{t\ell}\alpha^*_\ell - \sum_{j\in DC^t} w(j)\left(2\varepsilon R^2_j + \sum_{\ell\in IC^t\cup\{t\}} p_{tj\ell}(1+2\varepsilon)(2R_jd(j,\ell)+d^2(j,\ell))\right)\right] \\
&\qquad\qquad \leq \beta_t\cdot\left(\hat{f} + \sum_{\ell\in IC^t\cup[t:s]} \rho w(\ell)d^2(\ell,i) + \sum_{j\in DC^t} w(j)d^2(j,i)\right).
\end{aligned}$$

Summing over all $t\in[s]$, because of Observation 5.4, $\rho=2$, $\Phi=\sum_\ell w(\ell)d^2(j,\ell)$, and

$\sum_{t\in[s]}\beta_t \le 1$ (Corollary 5.18), we have

$$\sum_{\ell\in[s]} w(\ell)\alpha^*_\ell - \sum_{t\in[s]}\beta_t \cdot \sum_{j\in DC^t} w(j)\left(2\varepsilon R_j^2 + \sum_{\ell\in IC^t\cup\{t\}} p_{tj\ell}(1+2\varepsilon)(2R_j d(j,\ell) + d^2(j,\ell))\right)$$
$$\le \hat{f} + 2\Phi - \sum_{t\in[s]}\beta_t \sum_{j\in DC^t} w(j)d^2(j,i) = \hat{f} + 2\Phi - \sum_{j\in[s]} w(j)d_j^+ d^2(j,i)\ .$$

(Observation 5.6)

Therefore, we have

$$\sum_{j\in[s]} w(j)\cdot(\alpha^*_j - (\rho-1)R_j^2) - \hat{f} - 2\Phi \le -\sum_{j\in[s]} w(j)R_j^2 - \sum_{j\in[s]} w(j)d_j^+ d^2(j,i)$$
$$+ \sum_{t\in[s]}\beta_t \sum_{j\in DC^t} w(j)\left(2\varepsilon R_j^2 + \sum_{\ell\in IC^t\cup\{t\}} p_{tj\ell}(1+2\varepsilon)(2R_j d(j,\ell) + d^2(j,\ell))\right)$$
$$\le \sum_{j\in[s]} w(j)\left(-(1-2\varepsilon)R_j^2 - d_j^+ d^2(j,i) + \sum_{t\in[s]:j\in DC^t}\beta_t \sum_{\ell\in IC^t\cup\{t\}} p_{tj\ell}(1+2\varepsilon)(2R_j d(j,\ell) + d^2(j,\ell))\right)$$

($\sum_{t\in[s]:j\in DC^t}\beta_t = d_j^+ \le 1$, by Observation 5.6 and Corollary 5.19)

$$= \sum_{j\in[s]} w(j)\left(-(1-2\varepsilon)R_j^2 - d_j^+ d^2(j,i) + \sum_{\ell>j} P_{j\ell}(1+2\varepsilon)(2R_j d(j,\ell) + d^2(j,\ell))\right)$$

(definition of $P_{j\ell}$)

$$\le \sum_{j\in[s]} w(j)\left(-d_j^+ d^2(j,i) + (1+2\varepsilon)\sum_{\ell>j} P_{j\ell}d^2(j,\ell) + \frac{(1+2\varepsilon)^2}{1-2\varepsilon}\cdot\left(\sum_{\ell>j} P_{j\ell}d(j,\ell)\right)^2\right)$$

$$\le (1+2\varepsilon)\Phi + \sum_{j\in[s]} w(j)\cdot\left(-d_j^+ d^2(j,i) + \frac{(1+2\varepsilon)^2}{1-2\varepsilon}\cdot\left(\sum_{\ell>j} P_{j\ell}d(j,\ell)\right)^2\right) \qquad \text{(Lemma 5.8)}$$

$$\le (1+2\varepsilon)\Phi + \sum_{j\in[s]} w(j)\cdot\left(-d_j^+ d^2(j,i) + (1+7\varepsilon)\cdot\sum_{\ell>j} d_j^+ P_{j\ell}d^2(j,\ell)\right)$$

(ε is sufficiently small, Cauchy-Schwarz inequality)

$$\le (1+9\varepsilon)\Phi + \sum_{j<\ell} w(j)d_j^+ P_{j\ell}d^2(j,\ell) - \sum_{j\in[s]} w(j)d_j^+ d^2(j,i) \qquad (d_j^+ \le 1, \text{ Lemma 5.8})$$

$$\le (1+\ln(2)+9\varepsilon)\Phi\ . \qquad \text{(according to Subsection 5.3.4)}$$

C Missing Proofs in Section 6

C.1 Proof of Lemma 6.6

[AS:this part is not modified from the gpt writeup, and copied as is]

Define

$$Q(x) \stackrel{\text{def}}{=} 288x^4 - 235\tau x^3 - 1071x - 792 + 1128\tau.$$

Differentiating (27) gives

$$F'(x) = \frac{x}{2178}Q(x), \qquad Q'(x) = 1152x^3 - 705\tau x^2 - 1071, \qquad Q''(x) = x(3456x - 1410\tau). \tag{52}$$

We shall use the rational bounds

$$\frac{22894}{10000} < \tau < \frac{228943}{100000}, \qquad \Big(\frac{22894}{10000}\Big)^3 < 12 < \Big(\frac{228943}{100000}\Big)^3. \tag{53}$$

We split the interval into three pieces.

Step 1: the interval $[1, \frac{4}{3}]$. From (53),

$$3456 - 1410\tau > 3456 - 1410 \cdot \frac{228943}{100000} = \frac{2279037}{10000} > 0.$$

Hence $Q''(x) > 0$ on $[1, \tau]$, so Q' is increasing. Therefore on $[1, \frac{4}{3}]$,

$$Q'(x) \le Q'\Big(\frac{4}{3}\Big) = \frac{4979 - 3760\tau}{3} < \frac{4979 - 3760 \cdot (22894/10000)}{3} = -\frac{453643}{375} < 0.$$

Thus Q is strictly decreasing on $[1, \frac{4}{3}]$.

At the endpoints,

$$Q(1) = 893\tau - 1575 > 893 \cdot \frac{22894}{10000} - 1575 = \frac{2347171}{5000} > 0,$$

while

$$Q\Big(\frac{4}{3}\Big) = \frac{15416\tau - 35364}{27} < \frac{15416 \cdot (228943/100000) - 35364}{27} = -\frac{876839}{337500} < 0.$$

Therefore there is a unique $r \in (1, \frac{4}{3})$ such that $Q(r) = 0$. Since $F'(x)$ has the sign of $Q(x)$ by (52), the function F increases on $[1, r]$ and decreases on $[r, \frac{4}{3}]$. Hence

$$\max_{1 \le x \le 4/3} F(x) = F(r).$$

We now compare $F(r)$ with $F(\frac{4}{3})$. By the mean value theorem, there exists $\zeta \in (r, \frac{4}{3})$ such that

$$Q\Big(\frac{4}{3}\Big) - Q(r) = Q'(\zeta)\Big(\frac{4}{3} - r\Big).$$

Since $Q(r) = 0$ and $Q'(\zeta) \le -453643/375$, while

$$-Q\Big(\frac{4}{3}\Big) = \frac{35364 - 15416\tau}{27} < \frac{35364 - 15416 \cdot (22894/10000)}{27} = \frac{44131}{16875},$$

we get

$$\frac{4}{3} - r = \frac{-Q(\frac{4}{3})}{-Q'(\zeta)} < \frac{44131/16875}{453643/375}.$$

Because Q is decreasing on $[r, \frac{4}{3}]$ and vanishes at r, one has

$$0 \le -Q(x) < \frac{44131}{16875} \qquad (r \le x \le 4/3).$$

Therefore

$$\begin{aligned} 0 \le F(r) - F\Big(\frac{4}{3}\Big) &= \int_r^{4/3} -F'(x)\,dx = \int_r^{4/3} \frac{x(-Q(x))}{2178}\,dx \\ &\le \frac{4/3}{2178}\cdot\frac{44131}{16875}\cdot\Big(\frac{4}{3}-r\Big) \\ &< \frac{4/3}{2178}\cdot\frac{44131}{16875}\cdot\frac{44131/16875}{453643/375} \\ &= \frac{3895090322}{1125432995259375} < 4\times 10^{-6}. \end{aligned}$$

Next,

$$F\Big(\frac{4}{3}\Big) = \frac{97760\tau}{264627} + \frac{822427}{529254} < \frac{97760\cdot(228943/100000)}{264627} + \frac{822427}{529254} = \frac{793785221}{330783750} < 2.39971045.$$

Hence

$$\max_{1\le x\le 4/3} F(x) < 2.39971045 + 4\times 10^{-6} < \frac{12}{5}.$$

Step 2: the interval $[\frac{4}{3}, 2]$. Since $Q'' > 0$, the function Q is convex. We already know that $Q(\frac{4}{3}) < 0$. Also,

$$Q(2) = 1674 - 752\tau < 1674 - 752\cdot\frac{22894}{10000} = -\frac{29768}{625} < 0.$$

A convex function that is negative at both endpoints of an interval is negative throughout that interval, so $Q(x) < 0$ on $[\frac{4}{3}, 2]$. By (52), the function F is therefore strictly decreasing on $[\frac{4}{3}, 2]$, and hence

$$\max_{4/3\le x\le 2} F(x) = F\Big(\frac{4}{3}\Big) < \frac{12}{5}.$$

Step 3: the interval $[2, \tau]$. Again $Q'' > 0$, so Q' is increasing. Moreover,

$$Q'(2) = 8145 - 2820\tau > 8145 - 2820\cdot\frac{228943}{100000} = \frac{8444037}{5000} > 0.$$

Thus $Q'(x) > 0$ on $[2, \tau]$, so Q is increasing there. Since also

$$Q(2) = 1674 - 752\tau < 1674 - 752\cdot\frac{22894}{10000} = -\frac{29768}{625} < 0,$$

the function Q can cross zero at most once on $[2, \tau]$, and if it does, it crosses from negative to positive. Therefore $F'(x) = xQ(x)/2178$ can change sign at most once, from negative to positive, so the maximum of F on $[2, \tau]$ is attained at one of the endpoints.

At $x = 2$,

$$F(2) = \frac{752\tau + 3297}{2178} < \frac{752\cdot(228943/100000) + 3297}{2178} = \frac{31366571}{13612500} < \frac{12}{5}.$$

At $x = \tau$, using $\tau^3 = 12$ and $\tau^6 = 144$, we obtain

$$F(\tau) = \frac{48\tau^6 - 47\tau^6 - 357\tau^3 + (564\tau - 396)\tau^2 + 4665}{2178} = \frac{\tau^6 + 207\tau^3 - 396\tau^2 + 4665}{2178} = \frac{221}{66} - \frac{2}{11}\tau^2.$$

Hence, from the lower bound in (53),

$$F(\tau) < \frac{221}{66} - \frac{2}{11}\Big(\frac{22894}{10000}\Big)^2 = \frac{988148573}{412500000} < \frac{12}{5}.$$

So $F(x) < \frac{12}{5}$ throughout $[2, \tau]$ as well.

Combining the three steps proves the lemma.

C.2 Weighted: Full Proof

In this subsection, we prove Lemma 4.12. Define

$$a_t \stackrel{\text{def}}{=} d(t,i), \qquad W(S) \stackrel{\text{def}}{=} \sum_{j\in S} w(j) \quad (S \subseteq [s]), \qquad W_t \stackrel{\text{def}}{=} \sum_{u=t}^{s} w(u).$$

MC: minor quibble about this W notation: I keep reading it as a summation!

AS: i agree i switched it to (overloaded) W_t

As briefly discussed in the overview in Subsection 2.1, the proof has four steps. First, from the inequalities at time $\alpha_t - \varepsilon$ given in condition (3) of Lemma 4.12, we derive a family of constraints (P_t^w) by using the triangle inequality to lower bound the terms involving r_{jt}. Second, we choose coefficients $(\beta)_{t\in[s]}$ so that, in the weighted combination of these constraints, the coefficient of each α_t^* is exactly $w(t)$. Third, after collecting terms, we obtain a quadratic form in the distances to the facility i. After rescaling by $\sqrt{w(t)}$, the quadratic form is represented by an ordinary symmetric nonnegative matrix. Finally, we bound the top eigenvalue of the corresponding matrix by the Collatz–Wielandt Formula, which yields the claimed constant.

MC: I don't think it will be easy for the reader to map this to the steps in the earlier technical overview which specialized to the all DC case. We should probably have a brief overview in this section.[AS:done]

Step 1: From (I_t^w) to (P_t^w). For each $t \in [s]$, condition (3) gives

$$\sum_{j\in DC^t} w(j)(r_{jt}^2 - a_j^2) + \sum_{j\in IC^t} w(j)(\alpha_j^* - \rho a_j^2) + \sum_{j\geq t} w(j)(\alpha_t^* - \rho a_j^2) \leq \hat{f}. \tag{I_t^w}$$

MC: $(P_t^w)_{t\in[s]}$ and $(I_t^w)_{t\in[s]}$ are not yet defined.[AS:resolved]

Fix $t \in [s]$. For each $j \in DC^t$, conditions (1)–(2) and the triangle inequality imply

$$\alpha_t^* \leq (r_{jt} + d(j,t))^2 \leq (r_{jt} + a_j + a_t)^2 \leq r_{jt}^2 + (a_j + a_t)^2 + 2(a_j + a_t)R_j.$$

Rearranging and using $2a_j a_t \leq a_j^2 + a_t^2$, we obtain

$$r_{jt}^2 - a_j^2 \geq \alpha_t^* - 2a_t^2 - 3a_j^2 - 2(a_j + a_t)R_j.$$

Substituting into Equation (I_t^w), we obtain

$$\begin{aligned}(W(DC^t) + W_t)\alpha_t^* + \sum_{j\in IC^t} w(j)\alpha_j^* - \hat{f} \\ \leq 2W(DC^t)a_t^2 + \sum_{j\in DC^t} w(j)\big(3a_j^2 + 2(a_t + a_j)R_j\big) + \sum_{j\in IC^t} \rho w(j)a_j^2 + \sum_{j\geq t} \rho w(j)a_j^2.\end{aligned} \tag{P_t^w}$$

Step 2: Choice of the coefficients. We now choose nonnegative coefficients $\beta_1, \ldots, \beta_s$ and multiply each inequality (P_t^w) with β_t and sum over all $t \in [s]$ such that the coefficient of each α_t^* is exactly $w(t)$.

Lemma C.1. *For each $t \in [s]$, define*

$$M_t \stackrel{def}{=} W(DC^t), \qquad c_t \stackrel{def}{=} \frac{W_t + M_t}{w(t)}, \qquad g_t \stackrel{def}{=} \sum_{u\geq t} \beta_u,$$

$$\eta_t \stackrel{def}{=} \sum_{u>t:\, t\in DC^u} \beta_u, \qquad A_t \stackrel{def}{=} \frac{M_t\beta_t}{w(t)}.$$

Then there exist nonnegative coefficients $\beta_1, \ldots, \beta_s$ such that, for every $t \in [s]$,

$$\beta_t c_t + \sum_{u>t:\, t\in IC^u} \beta_u = 1, \tag{54}$$

and moreover,

$$0 \le \eta_t \le g_t, \qquad \sum_{t=1}^{s} \beta_t = 1, \qquad A_t + g_t \le 1.$$

Proof. Define the coefficients recursively, for $t = s, s-1, \dots, 1$, by

$$\beta_t = \frac{1 - \sum_{u>t:\, t \in IC^u} \beta_u}{c_t}. \tag{55}$$

Since Equation (54) is triangular, this is the unique solution of the system. We verify that the resulting coefficients are nonnegative.

Fix $t \in \{0\} \cup [s]$ and consider the suffix $\{\alpha^*_{t+1}, \dots, \alpha^*_s\}$. In the final weighted combination, its total coefficient is W_{t+1}, where by convention $W_{s+1} = 0$. On the other hand, for each $u > t$, the contribution of the uth inequality Equation (P_t^w) to this suffix is

$$W_u + W(DC^u) + W(IC^u \cap [t+1, u-1]).$$

Since IC^u and DC^u partition $[u-1]$, we have

$$\begin{aligned} &W_u + W(DC^u) + W(IC^u \cap [t+1, u-1]) \\ &\quad = W_u + W(DC^u) + W([t+1, u-1]) - W(DC^u \cap [t+1, u-1]) \\ &\quad = W_{t+1} + W(DC^u \cap [1, t]) \\ &\quad \ge W_{t+1} + M_t, \end{aligned}$$

where we used $DC^t \subseteq DC^u$ and $DC^t \subseteq [t-1]$. Summing over $u > t$ gives

$$(W_{t+1} + M_t) \sum_{u>t} \beta_u \le W_{t+1},$$

and hence

$$\sum_{u>t} \beta_u \le \frac{W_{t+1}}{W_{t+1} + M_t} \le 1.$$

Therefore,

$$\sum_{u>t:\, t \in IC^u} \beta_u \le \sum_{u>t} \beta_u \le 1,$$

so the numerator in Equation (55) is nonnegative, and hence $\beta_t \ge 0$.

The inequality $0 \le \eta_t \le g_t$ is immediate from the definitions.

To prove $\sum_{t=1}^{s} \beta_t = 1$, sum the coefficients of all α^*-variables in the weighted combination of Equation (P_t^w). Each inequality contributes total coefficient W_1, while by construction each α^*_u has coefficient $w(u)$. Hence

$$W_1 \sum_{t=1}^{s} \beta_t = W_1,$$

MC: Here, the W notation makes it hard to parse this equation.[AS:resolved]

so $\sum_{t=1}^{s} \beta_t = 1$, and in particular $g_t \le 1$.

Finally, fix $t \in [s]$ and consider the suffix $\{\alpha^*_t, \dots, \alpha^*_s\}$. Its total coefficient in the final weighted combination is W_t. For each $u \ge t$, the contribution of the uth inequality to this suffix is

$$W_u + W(DC^u) + W(IC^u \cap [t, u-1]).$$

Again using that IC^u and DC^u partition $[u-1]$, we obtain

$$\begin{aligned} &W_u + W(DC^u) + W(IC^u \cap [t, u-1]) \\ &\quad = W_u + W(DC^u) + W([t, u-1]) - W(DC^u \cap [t, u-1]) \\ &\quad = W_t + W(DC^u \cap [1, t-1]) \\ &\quad \geq W_t + M_t = c_t w(t). \end{aligned}$$

Summing over $u \geq t$ yields $c_t w(t) g_t \leq W_t$, so

$$g_t \leq \frac{W_t}{c_t w(t)}.$$

Since Equation (54) implies $c_t \beta_t \leq 1$, we also have $\beta_t \leq 1/c_t$. Therefore,

$$A_t + g_t = \frac{M_t \beta_t}{w(t)} + g_t \leq \frac{M_t}{c_t w(t)} + \frac{W_t}{c_t w(t)} = 1.$$

□

Step 3: The weighted quadratic form. We now compute the quadratic form.

Lemma C.2. *For each $t \in [s]$ and each $x \in \mathbb{R}^s$, define*

$$U_t(x) \stackrel{def}{=} \sum_{u:\, t \in DC^u} \beta_u x_u.$$

Then the weighted sum of all inequalities (P_t^w) over $t \in [s]$ implies

$$\sum_{t \in [s]} w(t)\big(\alpha_t^* - (\rho - 1) R_t^2\big) - \hat{f} \leq \rho \sum_{t \in [s]} w(t) a_t^2 + Q_\rho^w(a),$$

where, for every $x \in \mathbb{R}^s$,

$$Q_\rho^w(x) = \sum_{t \in [s]} w(t)\big(2A_t + (3 - \rho)\eta_t\big) x_t^2 + \frac{1}{\rho - 1} \sum_{t \in [s]} w(t)\big(U_t(x) + \eta_t x_t\big)^2. \tag{56}$$

Proof. By Equation (54), the weighted sum of the left-hand sides of $(P_t^w)_{t \in [s]}$ is $\sum_t w(t)\alpha_t^* - \hat{f}$. Let $\mathcal{R}$ denote the weighted sum of the right-hand sides. Then

$$\sum_{t \in [s]} w(t)\alpha_t^* - \hat{f} \leq \mathcal{R}.$$

We now collect the terms in $\mathcal{R}$.

First,

$$\sum_{t \in [s]} \beta_t \, 2W(DC^t) a_t^2 = \sum_{t \in [s]} 2w(t) A_t a_t^2.$$

Second,

$$\sum_{t \in [s]} \beta_t \sum_{j \in DC^t} 3w(j) a_j^2 = \sum_{t \in [s]} 3w(t) \eta_t a_t^2.$$

Third,

$$\sum_{t\in[s]}\beta_t\sum_{j\in DC^t}2w(j)(a_t+a_j)R_j=2\sum_{t\in[s]}w(t)R_t\big(U_t(a)+\eta_t a_t\big).$$

Finally, for each fixed t, the sets

$$\{u\in[s]: t\in IC^u\},\qquad \{u\in[s]: t\in DC^u\},\qquad \{u\in[s]: u\le t\}$$

partition $[s]$. Since $\sum_{u=1}^{s}\beta_u=1$ by Lemma C.1, it follows that

$$\sum_{u:\,t\in IC^u}\beta_u+\sum_{u\le t}\beta_u=1-\eta_t,$$

and hence

$$\sum_{u\in[s]}\beta_u\left(\sum_{j\in IC^u}\rho w(j)a_j^2+\sum_{j\ge u}\rho w(j)a_j^2\right)=\rho\sum_{t\in[s]}w(t)(1-\eta_t)a_t^2.$$

Therefore,

$$\mathcal{R}=\rho\sum_{t\in[s]}w(t)a_t^2+\sum_{t\in[s]}w(t)\big(2A_t+(3-\rho)\eta_t\big)a_t^2+2\sum_{t\in[s]}w(t)R_t\big(U_t(a)+\eta_t a_t\big).$$

Hence

$$\begin{aligned}&\sum_{t\in[s]}w(t)\big(\alpha_t^*-(\rho-1)R_t^2\big)-\hat f\\&\qquad\le\rho\sum_{t\in[s]}w(t)a_t^2+\sum_{t\in[s]}w(t)\big(2A_t+(3-\rho)\eta_t\big)a_t^2\\&\qquad\quad+\sum_{t\in[s]}w(t)\Big(2R_t\big(U_t(a)+\eta_t a_t\big)-(\rho-1)R_t^2\Big).\end{aligned}$$

As in the warm-up, we absorb the mixed term using

$$2Rx-(\rho-1)R^2\le\frac{x^2}{\rho-1}\qquad(R,x\in\mathbb{R}).$$

Applying this with $x=U_t(a)+\eta_t a_t$ for each t, we obtain exactly the claimed inequality. □

Step 4: Weighted matrix form. It remains to bound $Q_\rho^w(x)$ by a multiple of

$$\sum_{t\in[s]}w(t)x_t^2.$$

For $x\in\mathbb{R}^s$, write

$$\widetilde{x}\overset{\text{def}}{=}\big(\sqrt{w(1)}\,x_1,\dots,\sqrt{w(s)}\,x_s\big).$$

Lemma C.3. *Let B be the unique symmetric matrix such that*

$$Q_\rho^w(x)=\widetilde{x}^{\top}B\,\widetilde{x}\qquad\textit{for all } x\in\mathbb{R}^s.$$

Then, for every $t \in [s]$ and every $x \in \mathbb{R}^s_+$,

$$(B\widetilde{x})_t = \sqrt{w(t)}\left[\left(2A_t+(3-\rho)\eta_t+\frac{\eta_t^2}{\rho-1}\right)x_t+\frac{\eta_t}{\rho-1}U_t(x)+\frac{\beta_t}{(\rho-1)w(t)}\sum_{u\in DC^t} w(u)\big(U_u(x)+\eta_u x_u\big)\right].$$

Moreover, B is entrywise nonnegative for all $1 < \rho \le 3$. Consequently,

$$Q^w_\rho(x) \le \lambda_{\max}(B)\sum_{t\in[s]} w(t)x_t^2 \qquad \text{for all } x \in \mathbb{R}^s_+.$$

Proof. Since $Q^w_\rho(x) = \widetilde{x}^\top B\,\widetilde{x}$, we have

$$\frac{\partial Q^w_\rho(x)}{\partial x_t} = 2\sqrt{w(t)}\,(B\widetilde{x})_t.$$

Differentiating Equation (56) gives

$$\begin{aligned}\frac{(B\widetilde{x})_t}{\sqrt{w(t)}} &= \big(2A_t+(3-\rho)\eta_t\big)x_t+\frac{\eta_t}{\rho-1}\big(U_t(x)+\eta_t x_t\big)+\frac{\beta_t}{(\rho-1)w(t)}\sum_{u\in DC^t} w(u)\big(U_u(x)+\eta_u x_u\big)\\ &= \left(2A_t+(3-\rho)\eta_t+\frac{\eta_t^2}{\rho-1}\right)x_t+\frac{\eta_t}{\rho-1}U_t(x)+\frac{\beta_t}{(\rho-1)w(t)}\sum_{u\in DC^t} w(u)\big(U_u(x)+\eta_u x_u\big).\end{aligned}$$

Every coefficient in this expression is nonnegative when $1 < \rho \le 3$, so B is entrywise nonnegative. The final claim follows from the variational characterization of $\lambda_{\max}(B)$. □

We now fix $\rho = 5/2$ and replace η_t by g_t, since $\eta_t \le g_t$ by Lemma C.1. For every $t \in [s]$ and every $x \in \mathbb{R}^s_+$, Lemma C.3 gives

$$\frac{(B\widetilde{x})_t}{\sqrt{w(t)}} \le \left(2A_t+\frac{1}{2}g_t+\frac{2}{3}g_t^2\right)x_t+\frac{2}{3}g_tU_t(x)+\frac{2\beta_t}{3w(t)}\sum_{u\in DC^t} w(u)U_u(x)+\frac{2\beta_t}{3w(t)}\sum_{u\in DC^t} w(u)g_ux_u. \tag{57}$$

Lemma C.4. *Fix $\tau = \sqrt[3]{12}$. Then*

$$\lambda_{\max}(B) \le \max_{1\le x\le\tau} F(x),$$

where

$$F(x) = \frac{48x^6-47\tau x^5-357x^3+(564\tau-396)x^2+4665}{2178}. \tag{58}$$

Proof. Choose

$$h_t = (1+11g_t)^{-2/3} \qquad (t\in[s]), \qquad \widetilde{h}_t = \sqrt{w(t)}\,h_t.$$

Since B is symmetric and entrywise nonnegative, the Collatz–Wielandt Formula [Mey00, Theorem 8.3.3] for the spectral radius $\rho(B)$ of nonnegative matrix $B \ge 0$ gives

$$\lambda_{\max}(B) = \rho(B) = \min_{x\in\mathbb{R}^s_+}\max_{t\in[s],x_t\neq 0}\frac{(B\widetilde{x})_t}{\widetilde{x}_t} \le \max_{t\in[s]}\frac{(B\widetilde{h})_t}{\widetilde{h}_t}.$$

We claim that, for every $t \in [s]$,

$$\frac{2g_t U_t(h)}{3h_t} \le \frac{2}{11} g_t \Big((1+11g_t) - (1+11g_t)^{2/3} \Big), \tag{59}$$

$$\frac{2\beta_t}{3w(t)h_t} \sum_{u \in DC^t} w(u) g_u h_u \le A_t \cdot \frac{2}{3} (1+11g_t)^{2/3} \tau^{-2}, \tag{60}$$

$$\frac{2\beta_t}{3w(t)h_t} \sum_{u \in DC^t} w(u) U_u(h) \le A_t (1+11g_t)^{2/3} \cdot \frac{2}{11} (\tau - 1). \tag{61}$$

To prove Equation (59), note that the sets DC^u are monotone in u, so the set $\{u \in [s] : t \in DC^u\}$ is a suffix of $\{t+1, \dots, s\}$. Let u_t be its first element, with the convention $u_t = s+1$ if $t \notin DC^u$ for every u. Then

$$U_t(h) = \sum_{u \ge u_t} \beta_u h_u = \sum_{u \ge u_t} (g_u - g_{u+1})(1+11g_u)^{-2/3}.$$

Since the function $z \mapsto (1+11z)^{-2/3}$ is decreasing, this sum is bounded by the corresponding integral:

$$U_t(h) \le \int_0^{\eta_t} (1+11z)^{-2/3} \, dz \le \frac{3}{11} \Big((1+11\eta_t)^{1/3} - 1 \Big). \tag{62}$$

Together with $\eta_t \le g_t$, this gives Equation (59).

For Equation (60), the function $g \mapsto g(1+11g)^{-2/3}$ is increasing on $[0,1]$, so

$$g_u h_u = g_u (1+11g_u)^{-2/3} \le 12^{-2/3} = \tau^{-2}.$$

Hence

$$\sum_{u \in DC^t} w(u) g_u h_u \le W(DC^t) \tau^{-2} = M_t \tau^{-2},$$

and therefore

$$\frac{2\beta_t}{3w(t)h_t} \sum_{u \in DC^t} w(u) g_u h_u \le \frac{2\beta_t M_t}{3w(t)} (1+11g_t)^{2/3} \tau^{-2} = A_t \cdot \frac{2}{3} (1+11g_t)^{2/3} \tau^{-2}.$$

For Equation (61), Equation (62) implies

$$U_u(h) \le \frac{3}{11} \Big((1+11\eta_u)^{1/3} - 1 \Big) \le \frac{3}{11} (\tau - 1),$$

since $\eta_u \in [0,1]$. Therefore,

$$\frac{2\beta_t}{3w(t)h_t} \sum_{u \in DC^t} w(u) U_u(h) \le \frac{2\beta_t}{3w(t)} (1+11g_t)^{2/3} \, M_t \cdot \frac{3}{11} (\tau - 1) = A_t (1+11g_t)^{2/3} \cdot \frac{2}{11} (\tau - 1).$$

Dividing Equation (57) by h_t and combining Equations (59) to (61), we obtain

$$\begin{aligned}
\frac{(B\widetilde{h})_t}{\widetilde{h}_t} &\le \Big(2A_t + \frac{1}{2} g_t + \frac{2}{3} g_t^2 \Big) + \frac{2g_t U_t(h)}{3h_t} + \frac{2\beta_t}{3w(t)h_t} \sum_{u \in DC^t} w(u) U_u(h) + \frac{2\beta_t}{3w(t)h_t} \sum_{u \in DC^t} w(u) g_u h_u \\
&\le 2A_t + \frac{1}{2} g_t + \frac{2}{3} g_t^2 + \frac{2}{11} g_t \Big((1+11g_t) - (1+11g_t)^{2/3} \Big) \\
&\quad + A_t (1+11g_t)^{2/3} \cdot \frac{2}{11} (\tau - 1) + A_t \cdot \frac{2}{3} (1+11g_t)^{2/3} \tau^{-2} \\
&\le G(g_t),
\end{aligned}$$

where

$$G(g) \stackrel{\text{def}}{=} 2(1-g) + \frac{1}{2}g + \frac{2}{3}g^2 + \frac{2}{3}(1-g)(1+11g)^{2/3}\tau^{-2} + \frac{2}{11}(1-g)(1+11g)^{2/3}(\tau-1) + \frac{2}{11}g\Big((1+11g) - (1+11g)^{2/3}\Big). \tag{63}$$

The last inequality uses $A_t + g_t \le 1$ from Lemma C.1.

Substituting $g = (x^3-1)/11$ and $\tau^{-2} = \tau/12$ and simplifying yields

$$\frac{(B\widetilde{h})_t}{\widetilde{h}_t} \le G(g) = F(x) \qquad \text{for some } x \in [1,\tau].$$

This proves the claim. □

Since $F(x)$ is identical to its unweighted counterpart, from Lemma 6.6, we know that

$$\max_{x\in[1,\tau]} F(x) < \frac{12}{5}. \tag{64}$$

We can now finish the proof of Lemma 4.12. Using Lemma C.2 with $\rho = 5/2$, we obtain

$$\begin{aligned}
&\sum_{t\in[s]} w(t)\big(\alpha_t^* - (\rho-1)R_t^2\big) - \hat{f} \\
&\quad\le \frac{5}{2}\sum_{t\in[s]} w(t)a_t^2 + Q^w_{5/2}(a) && \text{(Lemma C.2)} \\
&\quad\le \frac{5}{2}\sum_{t\in[s]} w(t)a_t^2 + \lambda_{\max}(B)\sum_{t\in[s]} w(t)a_t^2 && \text{(Lemma C.3)} \\
&\quad\le \frac{5}{2}\sum_{t\in[s]} w(t)a_t^2 + \Big(\max_{x\in[1,\tau]} F(x)\Big)\sum_{t\in[s]} w(t)a_t^2 && \text{(Lemma C.4)} \\
&\quad\le \frac{5}{2}\sum_{t\in[s]} w(t)a_t^2 + \frac{12}{5}\sum_{t\in[s]} w(t)a_t^2 && \text{(Equation (64))} \\
&\quad= 4.9\sum_{t\in[s]} w(t)a_t^2.
\end{aligned}$$

This proves Lemma 4.12.