

Balanced Metric Labeling

(Extended Abstract)

Joseph (Seffi) Naor*

Roy Schwartz†

Abstract

We define the *balanced metric labeling* problem, a generalization of the metric labeling problem, in which each label has a *capacity*, i.e., at most ℓ vertices can be assigned to it. The balanced metric labeling problem is a powerful generalization of fundamental problems in the area of approximation algorithms, e.g., arrangements and balanced partitions of graphs. It is also motivated by resource limitations in certain practical scenarios. We focus on the case where the given metric is uniform and note that this case alone encompasses various well-known graph partitioning problems. We present the first (pseudo) approximation algorithm for this problem, achieving for any ε , $0 < \varepsilon < 1$, an approximation factor of $O\left(\frac{\ln n}{\varepsilon}\right)$, while assigning at most $\min\left\{\frac{O(\ln k)}{1-\varepsilon}, \ell + 1\right\} (1 + \varepsilon) \ell$ vertices to each label (k is the number of labels). Our approximation algorithm is based on a novel randomized rounding of a linear programming formulation that combines an embedding of the graph in a simplex together with spreading metrics and additional constraints that strengthen the formulation. Our randomized rounding technique uses both a randomized metric decomposition technique and a randomized label assignment technique. At the heart of our approach is the fact that only limited dependency is created between the labels assigned to different vertices, allowing us to bound the expected cost of the solution and the number of vertices assigned to each label, simultaneously. We note that the number of vertices assigned to each label is bounded via a new inequality of Janson [15] for tail bounds of (partly) dependent random variables.

*Computer Science Department, Technion, Haifa 32000, Israel. E-mail: naor@cs.technion.ac.il.

†Computer Science Department, Technion, Haifa 32000, Israel. E-mail: schwartz@cs.technion.ac.il.

1 Introduction

The *metric labeling* problem captures a wide variety of classification problems that arise in computer vision and other related areas. Typically, in a classification problem one wishes to assign labels to objects such that some measure of the quality of the labeling is optimized. Metric labeling captures many classification problems where the measure of the quality of the labeling depends not only on the cost of assigning a label to an object, but also on pairwise relations between the classified objects. Formally, the input for the metric labeling problem consists of an undirected graph $G = (V, E)$ with n vertices, a non-negative edge weight function $w : E \rightarrow \mathbb{R}^+$, a set $L = \{1, 2, \dots, k\}$ of k labels, a metric (L, d_L) over the set of labels and a non-negative assignment cost function $c : V \times L \rightarrow \mathbb{R}^+$. The goal is to find a labeling of the vertices $f : V \rightarrow L$ that minimizes: $\sum_{u \in V} c(u, f(u)) + \sum_{(u,v) \in E} w(u, v) \cdot d_L(f(u), f(v))$.

Kleinberg and Tardos [16] were the first to consider the metric labeling problem in its full generality. They presented a 2-approximation for uniform metrics and an $O(\log k \log \log k)$ -approximation for general metrics using the probabilistic tree embedding technique [4, 5]. This bound was recently improved to $O(\log k)$ [12]. We note that several special cases of the metric labeling problem received attention, e.g., [14, 7, 2]. Recently, Chuzhoy and Naor [8] showed that metric labeling is $\Omega(\log^{\frac{1}{2}-\delta} k)$ -hard to approximate under the assumption that $\text{NP} \not\subseteq \text{DTIME}(n^{\text{poly}(\log n)})$. Chekuri *et al.* [7] formulated a linear programming relaxation for metric labeling, where each vertex is associated with a distribution over the label set L , i.e., an embedding in a k -dimensional simplex, and distances are defined by an *earthmover* metric. Metric labeling also generalizes some classical and extensively studied combinatorial optimization problems. In the 0 -extension problem, there is a special set of terminals $\{t_1, t_2, \dots, t_k\}$, where each terminal t_i is a priori assigned label i . There is a metric on the label set, however, the assignment cost of labels to non-terminal vertices is zero. An $O(\log k)$ -approximation algorithm for the problem was given by [6] and later improved to an $O(\log k / \log \log k)$ -approximation [13]. The special case of a uniform metric is the well known *multiway cut* problem.

1.1 Balanced Metric Labeling

We introduce the *balanced metric labeling* problem in this paper. In this problem, each label has a *capacity*, ℓ , which is an upper bound on the number of vertices that can be assigned to it. The balanced metric labeling problem is a natural generalization of several well known graph partitioning and arrangement problems. For example, the b -balanced cut problem, $0 < b \leq 1/2$, asks for a minimum capacity cut $(S, V \setminus S)$ such that $b \cdot n \leq |S| \leq (1 - b) \cdot n$. Alternatively, we can say that each vertex can be assigned to one of two labels (at zero cost), indicating the side of the balanced cut to which the vertex belongs. Setting $\ell = (1 - b)n$, the resulting labeling is indeed a b -balanced cut. The metric d_L in this case is the uniform metric, and the cost of the labeling is the total weight of edges whose endpoints are in different sides of the cut. Similarly, the k -way balanced partitioning problem, where the goal is to partition a graph into k (almost) equal parts, is also a special case of the balanced metric labeling problem. Here, $|L| = k$ and the metric is uniform.

In the linear arrangement problem, the goal is to embed a graph onto the integral points of a straight line, while minimizing the sum of the lengths of the edges in the embedding. This is a special case of the balanced metric labeling, by setting $L = \{1, 2, \dots, n\}$, $\ell = 1$, and letting d_L be a linear metric. Thus, the resulting labeling defines a linear arrangement of the graph, and the cost of the labeling is the total length of the edges in the embedding. It is important to notice that *any* embedding of an undirected graph onto a finite metric space, where the goal is to minimize the volume of the embedding (total edge length in the embedding), is a special case of the balanced metric labeling problem. Thus, this problem encompasses various types of linear arrangements and embeddings of graphs onto grids and meshes, in any dimension.

An interesting application of our problem arises in the design of wireless networks. Consider, for example, a GSM network. A Base Transceiver Station (BTS) provides coverage to mobile users in a cell. The BTSs are further partitioned into clusters, where each cluster is controlled by a Base Station Controller (BSC). Handover, the maintenance of service to users moving between cells, is much simpler (or cheaper) within a

cluster. We assume that each BTS can only be assigned to a subset of the BSC-s, and a weighted graph is defined on the BTS-s, where edge weights correspond to the traffic between pairs of BTS-s. Thus, viewing the BSC-s as labels, where each BSC can control only a limited number of BTS-s, finding a minimum cost clustering of the BTS-s is an instance of the balanced metric labeling problem with a uniform metric and labeling costs belonging to $\{0, \infty\}$.

1.2 Our Results

We consider the balanced metric labeling problem with a uniform metric. As already mentioned, this case by itself is already a powerful generalization of many partitioning problems. We present the first (pseudo) approximation for this problem. For any $0 < \varepsilon < 1$, we find an $O\left(\frac{\ln n}{\varepsilon}\right)$ -approximation, such that there are at most $\min\left\{\frac{O(\ln k)}{1-\varepsilon}, \ell + 1\right\} (1 + \varepsilon) \ell$ vertices assigned to each label. We note that for $\ell = O(1)$ or $k = O(1)$ (which is the case in many partitioning problems), we obtain a constant multiplicative deviation in the number of vertices that receive the same label. Compare our result with the best known results for balanced graph partitioning: a pseudo-approximation was given by [17, 10], where the cost is approximated by an $O(\log n)$ factor, and the multiplicative deviation in the number of vertices assigned to a part (“label”) is a constant. Very recently, the $O(\log n)$ factor was improved to $O(\sqrt{\log n})$ [3]. A (true) $O(\log^2 n)$ -approximation algorithm for 2-way balanced cuts was given by [11]. However, it does not seem that any of the above results can be naturally extended to handle assignment costs of vertices to different parts (“labels”).

There are two main difficulties that we need to cope with in order to approximate the balanced uniform metric labeling problem. First, there is no obvious way how to bound the number of vertices assigned to a label in the approximation algorithms developed for the (uncapacitated) uniform metric labeling problem, e.g. [16]. Second, it is not clear how to incorporate label assignment costs with techniques developed for approximating partitioning problems, e.g., spreading metrics [10] and the technique of [3]. Metric decomposition techniques yield a partitioning of the metric into clusters with small radii, such that if the metric satisfies some spreading constraints, this usually implies that each cluster contains a small number of vertices. However, in general, there may not always exist a label that can be assigned to all vertices in a single cluster of the partition, for example, if some labels cannot be assigned to certain vertices (by setting the assignment cost to be ∞).

We formulate the balanced uniform metric labeling problem as a linear program combining the simplex embedding approach [7, 16] together with the spreading metric approach [9, 10]. The simplex embedding approach can be used to bound the assignment and edge costs, while the spreading metric approach is useful for bounding the number of vertices assigned to the same label. However, the combination of these two approaches is not enough and we strengthen the relaxation by adding a new constraint which can be viewed as a triangle inequality on the “closeness” of vertices (see Section 3 and constraint (6) in the LP relaxation).

Our approximation algorithm combines in a novel way randomized metric decomposition techniques and randomized label assignment techniques. This needs to be done in a way that balances the dependencies between the labels assigned to different vertices. If each vertex is assigned a random label independently, then it is easy to bound the number of vertices that are assigned to the same label, however, the expected cost of the labeling in this case can be unbounded (compared with the LP). If the labels chosen for the vertices are completely dependent (e.g., [16]), then it is easy to bound the cost of the solution, however, there is no way to bound the number of vertices that are assigned to the same label. Additionally, one can find examples where the algorithm of [16], in which all labels are dependent, assigns all vertices to the same label, even when adding capacity constraints to the relaxation of [16]. Therefore, it is a challenging problem to achieve a poly-logarithmic approximation factor while deviating as little as possible from the capacities.

Hence, we present a novel approximation technique that creates a delicate balance of dependencies, i.e., the label that a single vertex receives depends on only a limited number of other labels. This allows us to bound both the cost of the labeling and the number of vertices assigned to each label via a new inequality of Janson [15] for tail bounds of (partly) dependent random variables. We achieve a logarithmic approximation

factor and a low (i.e. logarithmic in k) multiplicative deviation in the number of vertices assigned to each label. We note that the deviation is larger than the deviation achieved for partitioning problems [17, 10, 3], since we have to cope with assignment costs in the range $[0, \infty)$, instead of a fixed assignment cost of zero.

2 Preliminaries

The input for the balanced uniform metric labeling problem consists of an undirected graph $G = (V, E)$ with n vertices, a non-negative edge weight function $w : E \rightarrow \mathbb{R}^+$, a set $L = \{1, 2, \dots, k\}$ of k labels, a non-negative assignment cost function $c : V \times L \rightarrow \mathbb{R}^+$, and a capacity parameter ℓ . The goal is to find a labeling of the vertices $f : V \rightarrow L$ that minimizes: $\sum_{u \in V} c(u, f(u)) + \sum_{(u,v) \in E} w(u, v) \cdot d_{unif}(f(u), f(v))$, such that $\forall j \in L, |\{u \mid f(u) = j\}| \leq \ell$. We denote the uniform metric by d_{unif} , i.e., $\forall i, j \in L, i \neq j, d_{unif}(i, j) = 1$, and $\forall i \in L, d_{unif}(i, i) = 0$.

We note that we can determine in polynomial time using flow techniques whether a feasible solution having finite cost exists for a given instance. We can assume that G is a complete graph by setting the weight of all missing edges to 0. We use the following density function (similarly to [4]).

Definition 1. $R \sim \text{Radius}(\varepsilon)$ if its density is: $f_R(r) = \left(\frac{n}{n-1}\right) \cdot \frac{1+\varepsilon}{\varepsilon} \ln n \cdot n^{-r \cdot \frac{1+\varepsilon}{\varepsilon}}, r \in \left[0, \frac{\varepsilon}{1+\varepsilon}\right]$.

Lemma 1. $\text{Radius}(\varepsilon)$ is a legal probability measure for any $\varepsilon > 0$.

Conclusion 2. For any $0 \leq r_1 \leq r_2 \leq \frac{\varepsilon}{1+\varepsilon}$,

$$\begin{aligned} \Pr[0 \leq R \leq r_1] &= \frac{n}{n-1} \cdot \left(1 - e^{-r_1 \frac{1+\varepsilon}{\varepsilon} \ln n}\right) \\ \Pr[r_1 \leq R \leq r_2] &= \frac{n}{n-1} \cdot \left(1 - e^{-(r_2-r_1) \frac{1+\varepsilon}{\varepsilon} \ln n}\right) \cdot e^{-r_1 \frac{1+\varepsilon}{\varepsilon} \ln n} \\ \Pr\left[r_2 \leq R \leq \frac{\varepsilon}{1+\varepsilon}\right] &= \frac{n}{n-1} \cdot \left(e^{-r_2 \frac{1+\varepsilon}{\varepsilon} \ln n} - \frac{1}{n}\right). \end{aligned}$$

3 The Relaxation

We formulate the balanced uniform metric labeling problem as a linear program that combines spreading metrics with an embedding in a k -dimensional simplex, in which distances are bounded from below by the variation distance (ℓ_1 -distance) together with additional new constraints.

In our formulation, each vertex is associated with a probability measure over the label set L , similarly to [7, 16]. Hence, vertex v is associated with variables $\varphi(v, j), 1 \leq j \leq k$. The *probability measure* of v over L is a vector $\varphi(v) = (\varphi(v, 1), \varphi(v, 2), \dots, \varphi(v, k))$ (see constraints (1)). Furthermore, we add *capacity* constraints (see constraints (2)) which guarantee that each label is not assigned to more than ℓ vertices.

We introduce a set of variables, $c_j(u, v), 1 \leq j \leq k, u, v \in V$, where $c_j(u, v)$ measures the ‘‘closeness’’ of $\varphi(u)$ and $\varphi(v)$ with respect to label j (see constraints (3)). Each edge (u, v) is also associated with a distance $d(u, v)$ which is related to the probability measures via the closeness variables (see constraints (4)). The variation distance of the points $\varphi(v), v \in V$, in the k -dimensional simplex is used to bound d from *below*. It is important to notice that d is a variable of the LP relaxation and not part of the input. Notice that constraints (1), (3) and (4) *alone* are equivalent to the relaxation of [16] for the uncapacitated uniform case.

The spreading constraints (see constraints (5)) imply that each sphere of bounded radius does not contain too many vertices. We strengthen the relaxation by adding a *new* constraint that can intuitively be seen as a ‘‘triangle inequality’’ constraint for the closeness variables. We call these constraints *closeness triangle inequality* (see constraints (6)). These constraints are important in bounding the separation cost of neighboring vertices and note that they can be made linear. Additionally, these constraints (with the variation distance constraints (4)) imply that (V, d) is a semi-metric (see Lemma 3). We refer to Appendix B for an example that constraint (6) is *stronger* than an explicit constraint stating that d satisfies the triangle inequality.

$$\begin{aligned}
\min \quad & \sum_{v \in V} \sum_{j \in L} c(v, j) \cdot \varphi(v, j) + \sum_{u, v \in V} w(u, v) \cdot d(u, v) \quad \text{s.t.} \\
& \sum_{j \in L} \varphi(v, j) = 1 \quad \forall v \in V \quad \text{(probability measure)} \quad (1) \\
& \sum_{v \in V} \varphi(v, j) \leq \ell \quad \forall j \in L \quad \text{(capacity)} \quad (2) \\
& c_j(u, v) \leq \varphi(u, j), \varphi(v, j) \quad \forall u, v \in V, \forall j \in L \quad \text{(closeness)} \quad (3) \\
& d(u, v) = 1 - \sum_{j \in L} c_j(u, v) \quad \forall u, v \in V \quad \text{(variation distance)} \quad (4) \\
& \sum_{v \in S} d(u, v) \geq |S| - \ell \quad \forall S \subseteq V, \forall u \in S \quad \text{(spreading)} \quad (5) \\
& \sum_{j \in L} |c_j(u, v) - c_j(u, w)| \leq 1 - \sum_{j \in L} c_j(v, w) \quad \forall u, v, w \in V \quad \text{(closeness } \Delta\text{-inequal.)} \quad (6) \\
& \varphi(v, j) \geq 0, c_j(u, v) \geq 0, d(u, v) \geq 0 \quad \forall u, v \in V, \forall j \in L
\end{aligned}$$

The proofs of the following lemmas appear in Appendices A, C and D.

Lemma 3. *For any feasible solution for the above program, (V, d) is a semi-metric.*

Lemma 4. *The value of an optimal solution for the above program is a lower bound on the cost of an optimal solution to the balanced uniform metric labeling problem.*

Lemma 5. *An optimal solution for the above program is computable in polynomial time.*

4 The Approximation Algorithm

Our novel technique for approximating the balanced uniform metric labeling problem combines randomized metric decomposition techniques with randomized label assignment techniques. We prove that our combined technique creates limited dependencies between the labels that vertices are assigned to. This enables us to bound both the cost of the solution and the number of vertices assigned to the same label, simultaneously.

Our approximation algorithm first assigns an initial labeling denoted by f^* , which we call a *root labeling*. This is done by a random procedure **RootLabel** and we defer its description to Section 5.2. We note that for every $v \in V$, $f^*(v)$ is a random variable satisfying the following property:

$$\Pr[f^*(v) = j] = \varphi(v, j) \quad , \quad \forall v \in V, \forall j \in L. \quad (7)$$

We do not assume that the random variables $\{f^*(v)\}_{v \in V}$ generated by **RootLabel** are necessarily independent. However, we assume that the random variables $\{f^*(v)\}_{v \in V}$ are independent of any other random choices made during our algorithm. We note that Property (7) is required for bounding the expected cost of the solution found by our approximation algorithm.

Our approximation algorithm, **ApproxLabel**, assigns labels to vertices in iterations. **ApproxLabel** conducts a single iteration for each vertex, where the vertex that corresponds to an iteration is called the *root* vertex of that iteration. We assume that the root vertices are determined in each iteration according to a prespecified ordering of the vertices $V = \{u_1, u_2, \dots, u_n\}$. This order is arbitrary and independent of any random choice made throughout the algorithm. In each iteration, the root label of that iteration's root vertex is assigned to a random (possibly empty) subset of vertices that are still unassigned to any label.

Let us consider iteration i with root vertex u_i . The random subset chosen in such an iteration is constructed from the intersection of two other random subsets. The first random subset consists of all vertices that belong to a sphere of radius R around u_i , where $R \sim \text{Radius}(\varepsilon)$. Hence, the first subset is

obtained via a randomized metric decomposition technique. The second subset consists of all vertices that are “close” to u_i with respect to the root label of u_i , i.e. $f^*(u_i)$. Vertex v is said to be “close” to vertex u_i with respect to label $f^*(u_i)$, if $c_{f^*(u_i)}(u_i, v) \geq \alpha$, where $\alpha \sim \text{Unif}[0, \varphi(u_i, f^*(u_i))]$. Recall that $c_{f^*(u_i)}(u_i, v) \leq \min\{\varphi(u_i, f^*(u_i)), \varphi(v, f^*(u_i))\}$ and the variable $c_{f^*(u_i)}(u_i, v)$ denotes the “closeness” of vertices v and u_i in terms of label $f^*(u_i)$. Hence, the second subset is obtained via a randomized label assignment technique. Thus, the random subset chosen in iteration i is the following intersection of two random subsets: $\{x \mid d(u_i, x) \leq R\} \cap \{x \mid c_{f^*(u_i)}(u_i, x) \geq \alpha\}$.

The subsets can be viewed in Figure 1. We assume that all random variables chosen by **ApproxLabel** are independent of each other. That is, for each iteration, the R 's and α 's are independent of each other and also independent of any other R 's and α 's chosen in any other iteration.

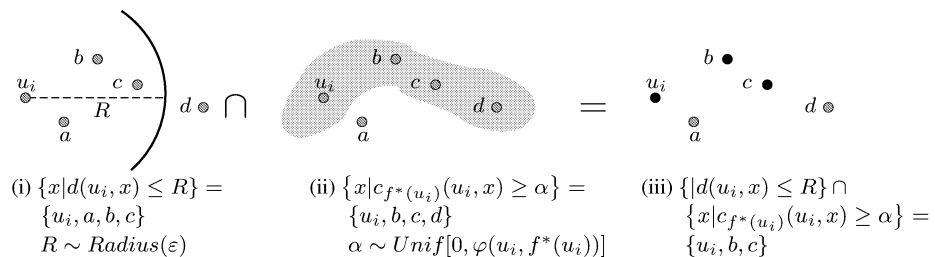


Figure 1: Iteration i with root u_i . Assume $f^*(u_i)$ is denoted by the black color.

We are now ready to state the approximation algorithm. We assume that an optimal solution for the linear program of the relaxation has been computed. At the end of each iteration we denote by V' the subset of vertices that are still not assigned to any label.

Algorithm ApproxLabel($\{d(u, v)\}_{u, v \in V}, \{\varphi(u)\}_{u \in V}, \{c_j(u, v)\}_{u, v \in V, j \in L}$)

1. $f^* \leftarrow \text{RootLabel}(\{\varphi(u)\}_{u \in V})$.
2. $V' \leftarrow V$.
3. For $i = 1$ to n do:
4. Choose $R \sim \text{Radius}(\varepsilon)$ and $\alpha \sim \text{Unif}[0, \varphi(u_i, f^*(u_i))]$.
5. For each $v \in V' \cap \{x \mid d(u_i, x) \leq R\} \cap \{x \mid c_{f^*(u_i)}(u_i, x) \geq \alpha\}$, set $f(v) \leftarrow f^*(u_i)$.
6. $V' \leftarrow V' \setminus (\{x \mid d(u_i, x) \leq R\} \cap \{x \mid c_{f^*(u_i)}(u_i, x) \geq \alpha\})$.
7. Output f .

Lemma 6 (See Appendix E for proof). *All vertices are labeled by ApproxLabel.*

5 Analysis of ApproxLabel

At the heart of the analysis of **ApproxLabel** is a proof that limited dependency between the labels exists, i.e., the label that a vertex receives depends on only a limited number of other labels. This allows us to simultaneously bound both the cost of the solution and the number of vertices assigned to each label. We show that the expected cost of the labeling is at most $O\left(\frac{\log n}{\varepsilon}\right)$ times the value of an optimal solution. We can prove this bound despite the lack of “full” dependency among the labels assigned to vertices (as in [16]). We note that proving this bound requires at least a limited level of dependency. We show that at most $\min\left\{\frac{O(\log k)}{1-\varepsilon}, \ell + 1\right\} (1+\varepsilon)\ell$ vertices are assigned to the same label. We are able to prove this bound, despite the lack of independence between the labels assigned to vertices, via a new inequality of Janson [15] for tail

bounds of (partly) dependent random variables.

5.1 Approximation Factor of ApproxLabel

We bound the expected cost of the solution found by **ApproxLabel** in two stages. First, we show that the separation cost of adjacent vertices is at most $O\left(\frac{\ln n}{\varepsilon}\right)$ times the separation cost of these vertices in the relaxation. Second, we show that the expected assignment cost of labels to vertices is at most $1 + \varepsilon$ times the assignment cost of labels to vertices in the relaxation. The general method used for proving these bounds is somewhat similar to a method used in [4], though in our case the analysis is much more involved, since each random subset chosen is constructed from two different random subsets. We use the following notations to simplify the presentation: $\bar{d}(u, v) = \min\left\{\frac{\varepsilon}{1+\varepsilon}, d(u, v)\right\}$, and $\tau = \frac{1+\varepsilon}{\varepsilon} \ln n$.

Expected Separation Cost of Neighboring Vertices. Define the following events for every $u, v \in V$: $A(u, v)$ - Vertices u and v are assigned to labels at different iterations; $B_i(u, v)$ - During iteration i exactly one of u and v is assigned to a label; and $C_i(u, v)$ - At the end of iteration i neither u nor v are assigned to any label. Consider an iteration i where u_i is the root and in which $\mathbb{P}[C_{i-1}(u, v)] > 0$, and assume without loss of generality that $\bar{d}(u_i, u) \leq \bar{d}(u_i, v)$. Therefore,

$$\begin{aligned} \mathbb{P}[B_i(u, v) \mid C_{i-1}(u, v)] &= \mathbb{P}[d(u_i, u) \leq R \leq d(u_i, v)] \cdot \mathbb{P}[c_{f^*(u_i)}(u_i, u) \geq \alpha] + \\ &\quad \mathbb{P}\left[\bar{d}(u_i, v) \leq R \leq \frac{\varepsilon}{1+\varepsilon}\right] \cdot \mathbb{P}[|\{u, v\} \cap \{x \mid c_{f^*(u_i)}(u_i, x) \geq \alpha\}| = 1] \\ \mathbb{P}[C_i(u, v) \mid C_{i-1}(u, v)] &= \mathbb{P}[0 \leq R \leq d(u_i, u)] + \\ &\quad \mathbb{P}[\bar{d}(u_i, u) \leq R \leq \bar{d}(u_i, v)] \cdot \mathbb{P}[c_{f^*(u_i)}(u_i, u) < \alpha] + \\ &\quad \mathbb{P}\left[\bar{d}(u_i, v) \leq R \leq \frac{\varepsilon}{1+\varepsilon}\right] \cdot \mathbb{P}[|\{u, v\} \cap \{x \mid c_{f^*(u_i)}(u_i, x) \geq \alpha\}| = 0]. \end{aligned}$$

The proof of the following lemma appears in Appendix F.

Lemma 7. For every iteration $1 \leq i \leq n$ and any pair of vertices $u, v \in V$:

$$(a) \mathbb{P}[c_{f^*(u_i)}(u_i, u) \geq \alpha] = 1 - d(u_i, u). \quad (b) \mathbb{P}[|\{u, v\} \cap \{x \mid c_{f^*(u_i)}(u_i, x) \geq \alpha\}| = 1] \leq d(u, v).$$

Clearly, for every $u, v \in V$, $\mathbb{P}[C_0(u, v)] = 1$, $\mathbb{P}[C_n(u, v)] = 0$ (since, at the end all vertices are labeled), and if there is an iteration i^* such that $\mathbb{P}[C_{i^*}(u, v)] = 0$, then for every iteration $i > i^*$: $\mathbb{P}[C_i(u, v)] = 0$. Define for every $u, v \in V$, $i^*(u, v) = \operatorname{argmin}_i \{\mathbb{P}[C_i(u, v)] = 0\}$. Obviously, the following reverse recursive formula on $\mathbb{P}[A(u, v) \mid C_{i-1}(u, v)]$ holds for every $1 \leq i \leq i^*(u, v) - 1$:

$$\begin{aligned} \mathbb{P}[A(u, v) \mid C_{i-1}(u, v)] &= \mathbb{P}[B_i(u, v) \mid C_{i-1}(u, v)] + \\ &\quad \mathbb{P}[C_i(u, v) \mid C_{i-1}(u, v)] \cdot \mathbb{P}[A(u, v) \mid C_{i-1}(u, v) \cap C_i(u, v)] = \\ &\quad \mathbb{P}[B_i(u, v) \mid C_{i-1}(u, v)] + \mathbb{P}[C_i(u, v) \mid C_{i-1}(u, v)] \cdot \mathbb{P}[A(u, v) \mid C_i(u, v)], \end{aligned}$$

where the last equality holds since $C_{i-1}(u, v) \cap C_i(u, v) = C_i(u, v)$.

The following lemma is crucial for the analysis of the approximation factor as it bounds the expected separation cost of adjacent vertices. We prove this lemma by reverse induction on the number of iterations, starting with iteration $i^*(u, v)$, and using the reverse recursive formula on $\mathbb{P}[A(u, v) \mid C_{i-1}(u, v)]$.

Lemma 8. $\mathbb{P}[A(u, v) \mid C_{i-1}(u, v)] \leq d(u, v) (1 + \varepsilon + \tau) \left(\frac{2n}{n-1} - \frac{i}{n-1}\right)$ for every $u, v \in V$, and every $1 \leq i \leq i^*(u, v)$.

Proof. We prove the lemma by reverse induction on the number of iterations.

Basis: Consider iteration $i^*(u, v)$. By the definition of $i^*(u, v)$, if at the beginning of this iteration neither u nor v are assigned to any label, then at its end at least one of them must be assigned to a label. Assuming without loss of generality that $d(u_{i^*(u,v)}, u) \leq d(u_{i^*(u,v)}, v)$, it must be the case that $d(u_{i^*(u,v)}, u) = 0$. If this is not the case, there is a positive probability that u is not assigned to a label in iteration $i^*(u, v)$. Since $d(u_{i^*(u,v)}, u) \leq d(u_{i^*(u,v)}, v)$, this implies that $\mathbb{P}\Pr[C_{i^*(u,v)}(u, v)] > 0$, contradicting the definition of $i^*(u, v)$.

Therefore, $\mathbb{P}\Pr[A(u, v) \mid C_{i^*(u,v)-1}] = \mathbb{P}\Pr[0 \leq R \leq \bar{d}(u_{i^*(u,v)}, v)] + \mathbb{P}\Pr[\bar{d}(u_{i^*(u,v)}, v) \leq R \leq \frac{\varepsilon}{\varepsilon+1}] \cdot \mathbb{P}\Pr[c_{f^*(u_{i^*(u,v)})(u_{i^*(u,v)}, v)} < \alpha]$. By Conclusion 2, by Lemma 7, by the fact that $1 - e^{-x} \leq x$ for every $x \geq 0$ and since $\bar{d}(u_{i^*(u,v)}, v) \leq d(u_{i^*(u,v)}, v)$, we obtain that:

$$\mathbb{P}\Pr[A(u, v) \mid C_{i^*(u,v)}(u, v)] \leq d(u_{i^*(u,v)}, v) \frac{n}{n-1} \left(\tau + e^{-\bar{d}(u_{i^*(u,v)}, v)\tau} - \frac{1}{n} \right) \leq d(u_{i^*(u,v)}, v) \frac{n}{n-1} (1 + \varepsilon + \tau).$$

Since $d(u_{i^*(u,v)}, u) = 0$, then $d(u_{i^*(u,v)}, v) = d(u, v)$ (d satisfies the triangle inequality, Lemma 3). Additionally, $i^*(u, v) \leq n$, hence: $\mathbb{P}\Pr[A(u, v) \mid C_{i^*(u,v)-1}(u, v)] \leq d(u, v) (1 + \varepsilon + \tau) \left(\frac{2n}{n-1} - \frac{i^*(u,v)}{n-1} \right)$.

Step: Assume correctness for $i + 1$, and assume without loss of generality that $\bar{d}(u_i, u) \leq \bar{d}(u_i, v)$. If $\bar{d}(u_i, u) = \frac{\varepsilon}{1+\varepsilon}$, then $\mathbb{P}\Pr[B_i(u, v) \mid C_{i-1}(u, v)] = 0$ and $\mathbb{P}\Pr[C_i(u, v) \mid C_{i-1}(u, v)] = 1$. Therefore, by the recursive formula we obtain that: $\mathbb{P}\Pr[A(u, v) \mid C_{i-1}(u, v)] = \mathbb{P}\Pr[A(u, v) \mid C_i(u, v)]$, and the lemma follows by the induction hypothesis for $i + 1$. Otherwise, $\bar{d}(u_i, u) < \frac{\varepsilon}{1+\varepsilon}$. By the induction hypothesis for $i + 1$ and the recursive formula:

$$\begin{aligned} \mathbb{P}\Pr[A(u, v) \mid C_{i-1}(u, v)] &\leq \mathbb{P}\Pr[B_i(u, v) \mid C_{i-1}(u, v)] + \\ &\quad \mathbb{P}\Pr[C_i(u, v) \mid C_{i-1}(u, v)] d(u, v) (1 + \varepsilon + \tau) \left(\frac{2n}{n-1} - \frac{i+1}{n-1} \right) \end{aligned}$$

By setting the values of $\mathbb{P}\Pr[B_i(u, v) \mid C_{i-1}(u, v)]$ and $\mathbb{P}\Pr[C_i(u, v) \mid C_{i-1}(u, v)]$, by Conclusion 2, by Lemma 7 and by the fact that $\mathbb{P}\Pr[\{u, v\} \cap \{x \mid c_{f^*(u_i)}(u_i, x) \geq \alpha\}] = 0 \leq \mathbb{P}\Pr[c_{f^*(u_i)}(u_i, u) < \alpha]$, we obtain that:

$$\begin{aligned} \mathbb{P}\Pr[A(u, v) \mid C_{i-1}(u, v)] &\leq \\ &\left(\frac{n}{n-1} \right) \left[\left(1 - e^{-(\bar{d}(u_i, v) - \bar{d}(u_i, u))\tau} \right) e^{-\bar{d}(u_i, u)\tau} (1 - d(u_i, u)) + \left(e^{-\bar{d}(u_i, v)\tau} - \frac{1}{n} \right) d(u, v) + \right. \\ &\left. \left[\left(1 - e^{-\bar{d}(u_i, u)\tau} \right) + \left(1 - e^{-(\bar{d}(u_i, v) - \bar{d}(u_i, u))\tau} \right) e^{-\bar{d}(u_i, u)\tau} d(u_i, u) + \left(e^{-\bar{d}(u_i, v)\tau} - \frac{1}{n} \right) d(u_i, u) \right] \cdot \right. \\ &\left. d(u, v) (1 + \varepsilon + \tau) \left(\frac{2n}{n-1} - \frac{i+1}{n-1} \right) \right] \end{aligned}$$

Since $1 - e^{-x} \leq x$, for every $x \geq 0$, and \bar{d} satisfies the triangle inequality (this is true since d satisfies the triangle inequality, Lemma 3), we obtain that:

$$\begin{aligned} \mathbb{P}\Pr[A(u, v) \mid C_{i-1}(u, v)] &\leq \frac{n}{n-1} \cdot \left[\bar{d}(u, v)\tau e^{-\bar{d}(u_i, u)\tau} \cdot (1 - d(u_i, u)) + \left(e^{-\bar{d}(u_i, v)\tau} - \frac{1}{n} \right) \cdot d(u, v) + \right. \\ &\left. \left[1 - e^{-\bar{d}(u_i, u)\tau} + d(u_i, u) \cdot \left(e^{-\bar{d}(u_i, u)\tau} - \frac{1}{n} \right) \right] \cdot d(u, v) (1 + \varepsilon + \tau) \left(\frac{2n}{n-1} - \frac{i+1}{n-1} \right) \right] \end{aligned}$$

Recall that $d(u_i, u) < \frac{\varepsilon}{1+\varepsilon}$, thus: $(1 + \varepsilon)(1 - d(u_i, u)) > 1$. Since $\bar{d}(u, v) \leq d(u, v)$, $\forall u, v \in V$, and $\bar{d}(u_i, u) \leq \bar{d}(u_i, v)$, we can conclude that:

$$\begin{aligned} \Pr[A(u, v) \mid C_{i-1}(u, v)] &\leq \frac{n}{n-1} d(u, v) (1 - d(u_i, u)) (\tau + 1 + \varepsilon) \left(e^{-\bar{d}(u_i, u)\tau} - \frac{1}{n} \right) + \\ &\quad \frac{d(u, v)}{n-1} \tau (1 - d(u_i, u)) + \left[\frac{n}{n-1} \left(1 - e^{-\bar{d}(u_i, u)\tau} \right) + \frac{n}{n-1} d(u_i, u) \left(e^{-\bar{d}(u_i, u)\tau} - \frac{1}{n} \right) \right] \cdot \\ &\quad d(u, v) (1 + \varepsilon + \tau) \frac{2n - i - 1}{n - 1} \end{aligned}$$

Since $\left[\frac{n}{n-1} \left(1 - e^{-\bar{d}(u_i, u)\tau} \right) + \frac{n}{n-1} d(u_i, u) \left(e^{-\bar{d}(u_i, u)\tau} - \frac{1}{n} \right) \right] \leq 1$ and $\tau(1 - d(u_i, u)) \leq (1 + \varepsilon + \tau)$:

$$\begin{aligned} \Pr[A(u, v) \mid C_{i-1}(u, v)] &\leq d(u, v) (\tau + 1 + \varepsilon) \cdot \\ &\quad \cdot \left[\frac{n}{n-1} \left(1 - e^{-\bar{d}(u_i, u)\tau} \right) + (d(u_i, u) + 1 - d(u_i, u)) \frac{n}{n-1} \left(e^{-\bar{d}(u_i, u)\tau} - \frac{1}{n} \right) \right] + \\ &\quad d(u, v) \left(\frac{n-i}{n-1} \right) (1 + \varepsilon + \tau) + \frac{d(u, v)}{n-1} (1 + \varepsilon + \tau) \end{aligned}$$

Notice that: $\left[\frac{n}{n-1} \left(1 - e^{-\bar{d}(u_i, u)\tau} \right) + (d(u_i, u) + 1 - d(u_i, u)) \frac{n}{n-1} \left(e^{-\bar{d}(u_i, u)\tau} - \frac{1}{n} \right) \right] = 1$. Therefore,

$$\begin{aligned} \Pr[A(u, v) \mid C_{i-1}(u, v)] &\leq d(u, v) (\tau + 1 + \varepsilon) \left[1 + \frac{n-i}{n-1} + \frac{1}{n-1} \right] \\ &= d(u, v) (\tau + 1 + \varepsilon) \left(\frac{2n}{n-1} - \frac{i}{n-1} \right) \end{aligned}$$

□

Expected Assignment Cost of Labels to Vertices. We now bound the assignment cost of labels to vertices. Define the following events for every $v \in V$ and $j \in L$: $A(v, j)$ - Vertex v is assigned to label j ; $B_i(v, j)$ - v is assigned to label j during iteration i ; and $C_i(v)$ - v is not assigned to any label at the end of iteration i . Consider an iteration i where $\Pr[C_{i-1}(v)] > 0$. Then,

$$\begin{aligned} \Pr[B_i(v, j) \mid C_{i-1}(v)] &= \Pr \left[\bar{d}(u_i, v) \leq R \leq \frac{\varepsilon}{1+\varepsilon} \right] \cdot \Pr \left[(c_{f^*(u_i)}(u_i, v) \geq \alpha) \wedge (f^*(u_i) = j) \right] \\ \Pr[C_i(v) \mid C_{i-1}(v)] &= \Pr \left[0 \leq R \leq \bar{d}(u_i, v) \right] + \Pr \left[\bar{d}(u_i, v) \leq R \leq \frac{\varepsilon}{1+\varepsilon} \right] \cdot \Pr \left[c_{f^*(u_i)}(u_i, v) < \alpha \right]. \end{aligned}$$

The proof of the following lemma appears in Appendix G.

Lemma 9. For every iteration $1 \leq i \leq n$, every label $j \in L$, and every vertex $v \in V$:

$$\Pr \left[(c_{f^*(u_i)}(u_i, v) \geq \alpha) \wedge (f^*(u_i) = j) \right] = c_j(u_i, v).$$

Clearly, for every $v \in V$, $\Pr[C_0(v)] = 1$, $\Pr[C_n(v)] = 0$ (since, at the end all vertices are labeled), and if there is an iteration i^* such that $\Pr[C_{i^*}(v)] = 0$, then for every iteration $i > i^*$: $\Pr[C_i(v)] = 0$. Define

for every $v \in V$, $i^*(v) = \operatorname{argmin}_i \{\mathbb{P}\mathbb{r}[C_i(v)] = 0\}$. Obviously, the following reverse recursive formula on $\mathbb{P}\mathbb{r}[A(v, j) \mid C_{i-1}(v)]$ holds for every $1 \leq i \leq i^*(v) - 1$:

$$\begin{aligned} \mathbb{P}\mathbb{r}[A(v, j) \mid C_{i-1}(v)] &= \mathbb{P}\mathbb{r}[B_i(v, j) \mid C_{i-1}(v)] + \\ &\quad \mathbb{P}\mathbb{r}[C_i(v) \mid C_{i-1}(v)] \cdot \mathbb{P}\mathbb{r}[A(v, j) \mid C_{i-1}(v) \cap C_i(v)] \\ &= \mathbb{P}\mathbb{r}[B_i(v, j) \mid C_{i-1}(v)] + \mathbb{P}\mathbb{r}[C_i(v) \mid C_{i-1}(v)] \cdot \mathbb{P}\mathbb{r}[A(v, j) \mid C_i(v)], \end{aligned}$$

where the last equality holds since $C_{i-1}(v) \cap C_i(v) = C_i(v)$.

The following lemma is crucial for the analysis of the approximation factor as it bounds the expected assignment cost of labels to vertices. We prove this lemma by reverse induction on the number of iterations, starting with iteration $i^*(v)$, and using the reverse recursive formula on $\mathbb{P}\mathbb{r}[A(v, j) \mid C_{i-1}(v)]$.

Lemma 10. *For every $v \in V$, every $1 \leq i \leq i^*(v)$, and every $j \in L$, $\mathbb{P}\mathbb{r}[A(v, j) \mid C_{i-1}(v)] \leq (1 + \varepsilon)\varphi(v, j)$.*

Proof. We prove the lemma by reverse induction on the number of iterations.

Basis: Consider iteration $i^*(v)$. If v is still not assigned to a label at the beginning of this iteration, then v must be assigned to a label during this iteration. Thus, $d(u_{i^*(v)}, v) = 0$ and $c_{j'}(u_{i^*(v)}, v) = \varphi(u_{i^*(v)}, j') = \varphi(v, j')$ for every $j' \in L$. If this is not the case, then there is a positive probability that v is not assigned to a label in iteration $i^*(v)$, implying that $\mathbb{P}\mathbb{r}[C_{i^*(v)}] > 0$, contradicting the definition of $i^*(v)$. Hence, $\mathbb{P}\mathbb{r}[A(v, j) \mid C_{i^*(v)-1}] = \varphi(u_{i^*(v)}, j) = \varphi(v, j) \leq (1 + \varepsilon)\varphi(v, j)$.

Step: Assume correctness for $i+1$. If $\bar{d}(u_i, v) = \frac{\varepsilon}{1+\varepsilon}$, then $\mathbb{P}\mathbb{r}[B_i(v, j) \mid C_{i-1}(v)] = 0$ and $\mathbb{P}\mathbb{r}[C_i(v) \mid C_{i-1}(v)] = 1$. Therefore, by the recursive formula we obtain that: $\mathbb{P}\mathbb{r}[A(v, j) \mid C_{i-1}(v)] = \mathbb{P}\mathbb{r}[A(v, j) \mid C_i(v)]$, and the lemma follows by the induction hypothesis for $i+1$. Otherwise, $d(u_i, v) < \frac{\varepsilon}{1+\varepsilon}$. By the induction hypothesis for $i+1$ and the recursive formula: $\mathbb{P}\mathbb{r}[A(v, j) \mid C_{i-1}(v)] \leq \mathbb{P}\mathbb{r}[B_i(v, j) \mid C_{i-1}(v)] + \mathbb{P}\mathbb{r}[C_i(v) \mid C_{i-1}(v)] \cdot (1 + \varepsilon)\varphi(v, j)$. By setting the values of $\mathbb{P}\mathbb{r}[B_i(v, j) \mid C_{i-1}(v)]$ and $\mathbb{P}\mathbb{r}[C_i(v) \mid C_{i-1}(v)]$, by Lemma 9, and by Lemma 7 we obtain that:

$$\begin{aligned} \mathbb{P}\mathbb{r}[A(v, j) \mid C_{i-1}(v)] &\leq \mathbb{P}\mathbb{r}\left[\bar{d}(u_i, v) \leq R \leq \frac{\varepsilon}{1+\varepsilon}\right] \cdot c_j(u_i, v) + \\ &\quad \left[\mathbb{P}\mathbb{r}[0 \leq R \leq \bar{d}(u_i, v)] + \mathbb{P}\mathbb{r}\left[\bar{d}(u_i, v) \leq R \leq \frac{\varepsilon}{1+\varepsilon}\right] d(u_i, v)\right] (1 + \varepsilon)\varphi(v, j) \end{aligned}$$

Recall that $d(u_i, v) < \frac{\varepsilon}{1+\varepsilon}$, thus, $(1 + \varepsilon)(1 - d(u_i, v)) > 1$. By constraint (3) (closeness constraint), $c_j(u_i, v) \leq \varphi(v, j)$. Therefore, we conclude:

$$\begin{aligned} \mathbb{P}\mathbb{r}[A(v, j) \mid C_{i-1}(v)] &\leq \mathbb{P}\mathbb{r}\left[\bar{d}(u_i, v) \leq R \leq \frac{\varepsilon}{1+\varepsilon}\right] \cdot (1 + \varepsilon)(1 - d(u_i, v)) \cdot \varphi(v, j) + \\ &\quad \left[\mathbb{P}\mathbb{r}[0 \leq R \leq \bar{d}(u_i, v)] + \mathbb{P}\mathbb{r}\left[\bar{d}(u_i, v) \leq R \leq \frac{\varepsilon}{1+\varepsilon}\right] d(u_i, v)\right] \cdot (1 + \varepsilon)\varphi(v, j) = (1 + \varepsilon)\varphi(v, j) \end{aligned}$$

□

Denote by OPT the cost of an optimal labeling of the balanced uniform metric labeling problem. Using Lemmas 4, 8 and 10 we complete the proof of the approximation factor of **ApproxLabel** (refer to Appendix H for the proof):

Theorem 11. *The expected cost of a solution of **ApproxLabel** is at most: $\left(\frac{1+\varepsilon}{\varepsilon} \ln n + 1 + \varepsilon\right) \left(\frac{2n}{n-1}\right) \cdot OPT$.*

5.2 Bounding The Number of Vertices Assigned to a Label

We present two upper bounds on the number of vertices assigned to a single label. Each upper bound corresponds to a different **RootLabel** procedure that satisfies Property (7). Neither of the two upper bounds is strictly better than the other, thus one can take the better of the two bounds for each instance.

As stated at the beginning of the paper, spreading constraints play a crucial role in the proof of an upper bound on the number of vertices that are assigned to the same label. We need the following lemma for both upper bounds, since it gives a lower bound on the diameter of a “large” subset of vertices. The proof of the following lemma appears in Appendix I.

Lemma 12. *Assume (V, d) is a semi-metric that satisfies the spreading constraints (5). Then, for every $S \subseteq V$ such that $|S| > \gamma\ell$ ($\gamma > 1$) and every $u \in S$, there exists a vertex $v \in S$ such that $d(u, v) > 1 - \frac{1}{\gamma}$.*

In order to prove the two upper bounds, we need to use the fact that spheres with a bounded diameter do not contain too many vertices. This can be easily derived from the previous lemma.

Corollary 13. *For every $v \in V$ and every $\gamma > 1$, $\left| \left\{ x \mid d(v, x) \leq 1 - \frac{1}{\gamma} \right\} \right| \leq \gamma\ell$.*

5.2.1 Upper Bound of $\frac{O(\ln k)}{1-\varepsilon}(1+\varepsilon)\ell$

In order to achieve this upper bound, we use the following simple **RootLabel** procedure, that assigns (independently) to each vertex a label according to its distribution over the label set. The random variable β is chosen independently in each iteration.

Algorithm RootLabel($\{\varphi(u)\}_{u \in V}$)

1. for $i = 1$ to n do:
2. Choose $\beta \sim \text{Unif}[0, 1]$.
3. Set $f^*(u_i) \leftarrow \min\{j > 0 \mid \sum_{j'=1}^j \varphi(u_i, j') \geq \beta\}$.
4. Output f^* .

Clearly, Property (7) is maintained in **RootLabel**. Define for every $v \in V$ and every $j \in L$ an indicator $Y_{v,j}$ for the event that at the end of algorithm **ApproxLabel**, vertex v is assigned to label j . According to Lemma 10, $\mathbb{E}[Y_{v,j}] \leq (1+\varepsilon)\varphi(v, j)$. By constraints (2) (capacity constraints), we obtain that $\mathbb{E}[\sum_{v \in V} Y_{v,j}] \leq (1+\varepsilon)\ell$.

If $\{Y_{v,j}\}_{v \in V}$ were independent random variables for every j , a simple Chernoff bound would suffice to achieve a good bound on the number of vertices assigned to each label. However, $\{Y_{v,j}\}_{v \in V}$ are clearly dependent. Thus, we use a more delicate bound by showing that $\{Y_{v,j}\}_{v \in V}$ are only “partly” dependent. This is done by showing that the maximum degree in the dependency graph of $\{Y_{v,j}\}_{v \in V}$ is not too “large”. The proof of the following lemma appears in Appendix J.

Lemma 14. *For every $v \in V$ and every $j \in L$, $Y_{v,j}$ is independent of $\{Y_{u,j} \mid d(u, v) > 2\frac{\varepsilon}{1+\varepsilon}\}$.*

We use the following definition of Janson [15] for a *dependency graph* for the random variables $\{Y_{v,j}\}_{v \in V}$, for every $j \in L$. In fact, this definition is equivalent to the one used in the Lovász Local Lemma [1].

Definition 2. *A graph $\Gamma = (V', E')$ is a dependency graph for $\{Y_{v,j}\}_{v \in V}$, if $V' = \{Y_{v,j}\}_{v \in V}$ and for every $V'' \subset V'$ and $Y_{v,j} \notin V''$ such that $Y_{v,j}$ is not connected by any edge in E' to vertices in V'' , $Y_{v,j}$ is independent of $\{Y_{v,j}\}_{v \in V''}$.*

Let us define a graph as follows: $\Gamma_j = \left(\{Y_{v,j}\}_{v \in V}, \left\{ (Y_{u,j}, Y_{v,j}) \mid d(u, v) \leq 2\frac{\varepsilon}{1+\varepsilon} \right\} \right)$. Clearly, by the above definition of a dependency graph and by Lemma 14, Γ_j is a dependency graph for $\{Y_{v,j}\}_{v \in V}$. Recall that $\varepsilon < 1$, hence by Corollary 13, the maximum degree of Γ_j is at most $\frac{(1+\varepsilon)\ell}{1-\varepsilon} - 1$. Let us denote the

maximum degree of Γ_j by Δ_j . Janson [15] proved the following inequality for every $t > 0$:

$$\mathbb{P}_r \left[\sum_{v \in V} Y_{v,j} \geq \mathbb{E} \left[\sum_{v \in V} Y_{v,j} \right] + t \right] \leq \exp \left(- \frac{8t^2}{25(\Delta_j + 1) \left(\sum_{v \in V} \text{Var}(Y_{v,j}) + \frac{bt}{3} \right)} \right),$$

where b satisfies the following condition: $Y_{v,j} - \mathbb{E}[Y_{v,j}] \leq b$, for every $v \in V$. Since for every $v \in V$, $Y_{v,j}$ is an indicator and $\mathbb{E} \left[\sum_{v \in V} Y_{v,j} \right] \leq (1 + \varepsilon)\ell$, we get that $b = 1$ and $\sum_{v \in V} \text{Var}(Y_{v,j}) \leq (1 + \varepsilon)\ell$. Hence, we get that if we choose $t = \delta(1 + \varepsilon)\ell$, the following holds:

$$\mathbb{P}_r \left[\sum_{v \in V} Y_{v,j} \geq (1 + \delta)(1 + \varepsilon)\ell \right] \leq \exp \left(- \frac{8(1 - \varepsilon)\delta^2}{25 \left(1 + \frac{\delta}{3} \right)} \right).$$

We need the above probability to be at most $\frac{1}{2k}$. Therefore, it suffices to choose $\delta = \frac{O(\ln k)}{1 - \varepsilon}$. Using the union bound over all k labels, we can conclude that with probability at least $\frac{1}{2}$, there are at most $(1 + \delta)(1 + \varepsilon)\ell = \frac{O(\ln k)}{1 - \varepsilon}(1 + \varepsilon)\ell$ vertices assigned to each label.

5.2.2 Upper Bound of $(\ell + 1)(1 + \varepsilon)\ell$

In order to achieve this bound we randomly choose root labels such that Property (7) is satisfied and there are at most $\ell + 1$ vertices having the same root label. This is done by sampling a random permutation matrix from a Birkhoff decomposition of a modified matrix constructed from $\{\varphi(v)\}_{v \in V}$. Full description appears in Appendix K.

The following theorem states the upper bound on the number of vertices that **ApproxLabel** assigns to each label. It follows directly by choosing the best upper bound from the two presented.

Theorem 15. *ApproxLabel finds a solution in which there are at most: $\min \left\{ \frac{O(\ln k)}{1 - \varepsilon}, \ell + 1 \right\} (1 + \varepsilon)\ell$ vertices assigned to each label, with probability of at least $\frac{1}{2}$.*

Acknowledgments

We wish to thank Moses Charikar, Julia Chuzhoy, and Danny Raz for helpful discussions.

References

- [1] N. Alon and J. H. Spencer, **The probabilistic method**, Second Edition, Wiley, New York, 2000.
- [2] A. Archer, J. Fakcharoenphol, C. Harrelson, R. Krauthgamer, and E. Tardos. Approximate classification via earthmover metrics, *Proceedings of the 15th ACM-SIAM Annual Symposium on Discrete Algorithms*, pp. 1072-1080, 2004.
- [3] S. Arora, S. Rao, and U. Vazirani. Expander flows and a $\sqrt{\log n}$ -approximation to sparsest cut, *Proceedings of the 36th ACM Symposium on Theory of Computing*, pp. 222-231, 2004.
- [4] Y. Bartal. Probabilistic approximation of metric spaces and its algorithmic applications, *Proceedings of the 37th IEEE Symposium on Foundations of Computer Science*, pp. 184-193, 1996.
- [5] Y. Bartal. On approximating arbitrary metrics by tree metrics, *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, (1998).
- [6] G. Călinescu, H. Karloff, and Y. Rabani. Approximation algorithms for the 0-extension problem, *Proceedings of the 12th ACM-SIAM Annual Symposium on Discrete Algorithms*, pp. 8–16, 2001.
- [7] C. Chekuri, S. Khanna, J. Naor, and L. Zosin. Approximation algorithms for the metric labeling problem via a new linear programming formulation. *Proceedings of the 12th Annual ACM/SIAM Symposium on Discrete Algorithms*, pp. 109–118, 2001.
- [8] J. Chuzhoy and J. Naor. The hardness of metric labeling, *Proceedings of the 45th IEEE Symposium on Foundations of Computer Science*, pp. 108–114, 2004.
- [9] G. Even, J. Naor, S. Rao and B. Schieber, Divide-and-conquer approximation algorithms via spreading metrics, *Journal of the ACM*, Vol. 47 (2000), pp. 585-616.
- [10] G. Even, J. Naor, S. Rao and B. Schieber, Fast approximate graph partitioning algorithms, *SIAM Journal on Computing*, Vol. 28 (1999), pp. 2187-2214.
- [11] U. Feige and R. Krauthgamer. A polylogarithmic approximation of the minimum bisection, *SIAM Journal on Computing*, Vol. 31 (2002), pp. 1090–1118.
- [12] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics, *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pp. 448-455, 2003.
- [13] J. Fakcharoenphol, C. Harrelson, S. Rao, and K. Talwar. An improved approximation algorithm for the 0-extension problem, *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 257–265, 2003.
- [14] A. Gupta and E. Tardos. Constant factor approximation algorithms for a class of classification problems. *Proceedings of the 32nd Annual ACM Symposium on the Theory of Computing*, pages 652–658, 2000.
- [15] S. Janson. Large deviations for sums of partly dependent random variables, *Random Structures and Algorithms*, Vol. 24, No. 3 (2004), pp.234–248.

- [16] J. Kleinberg and E. Tardos. Approximation algorithms for classification problems with pairwise relationships: metric labeling and Markov random fields, *Journal of the ACM*, Vol. 49, pp. 616-630, 2002.
- [17] T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM*, Vol. 46, pp. 787-832, 1999.

Appendix

A Proof of Lemma 3

Lemma 3. *For any feasible solution for the above program, (V, d) is a semi-metric.*

Proof. For every $u, v \in V$, $d(u, v) \geq 0$. Thus, we only need to prove the triangle inequality. For every $u, v, w \in V$, according to constraint (6) (closeness Δ -inequality):

$$\sum_{j \in L} |c_j(u, w) - c_j(u, v)| \leq 1 - \sum_{j \in L} c_j(w, v).$$

Removing the absolute value from the left hand side of the equation can only decrease the value of the left side, therefore:

$$\sum_{j \in L} (c_j(u, w) - c_j(u, v)) \leq 1 - \sum_{j \in L} c_j(w, v).$$

Using simple arithmetic we receive that:

$$1 - \sum_{j \in L} c_j(u, v) \leq 1 - \sum_{j \in L} c_j(u, w) + 1 - \sum_{j \in L} c_j(w, v).$$

Applying constraint (4) (variation distance), finishes the proof:

$$d(u, v) \leq d(u, w) + d(w, v).$$

□

B Closeness Triangle Inequality is Stronger than Triangle Inequality for d

From the proof of Lemma 3 we know that by using closeness Δ -inequality constraints (constraint (6)) with the variation distance constraints (constraint (4)), the triangle inequality for d can be proved. An interesting question is whether we can prove the opposite? (is it possible to prove that any solution for the LP relaxation that satisfies the triangle inequality constraint for d with all other constraints in the LP relaxation excluding constraint (6), also satisfies the closeness Δ -inequality).

The answer to this question is negative as we give an example in which the triangle inequality for d is satisfied, yet the closeness Δ -inequality is not satisfied. Consider a graph with only 3 vertices u, v and w , 2 labels and capacity $\ell = 2$. Set the following distributions on 2 labels: $\varphi(u) = (\frac{1}{2}, \frac{1}{2})$, $\varphi(v) = (\frac{1}{2} + \varepsilon, \frac{1}{2} - \varepsilon)$ and $\varphi(w) = (\frac{1}{2} - \varepsilon, \frac{1}{2} + \varepsilon)$, for some $\varepsilon > 0$. Set the closeness variables in the following way: $c_1(u, v) = \frac{1}{2}$, $c_2(u, v) = \frac{1}{2} - 2\varepsilon$, $c_1(u, w) = \frac{1}{2} - 2\varepsilon$, $c_2(u, w) = \frac{1}{2}$, $c_1(v, w) = \frac{1}{2} - \varepsilon$ and $c_2(v, w) = \frac{1}{2} - \varepsilon$. It can be verified that by choosing $\varepsilon = \frac{1}{4}$ all constraints of the LP relaxation, excluding the closeness Δ -inequality,

are satisfied. According to constraint (4) (variation distance), $d(u, v) = 2\varepsilon$, $d(u, w) = 2\varepsilon$ and $d(v, w) = 2\varepsilon$. Clearly, the triangle inequality for d is satisfied. However, the closeness Δ -inequality is not satisfied, since:

$$\sum_{j=1}^2 |c_j(u, v) - c_j(u, w)| = 4\varepsilon > 2\varepsilon = 1 - \sum_{j=1}^2 c_j(v, w) = d(v, w).$$

This implies that the closeness Δ -inequality is stronger than the triangle inequality for d .

C Proof of Lemma 4

Lemma 4. *The value of an optimal solution for the above program is a lower bound on the cost of an optimal solution to the balanced uniform metric labeling problem.*

Proof. Given an instance of the balanced uniform metric labeling problem, we show that it defines a feasible solution to the linear program, such that the objective function value of the linear program is equal to the cost of the labeling f .

Let us define a solution as follows. For each vertex v , set $\varphi(v, f(v)) = 1$ and $\varphi(v, j) = 0$ for every $j \neq f(v)$. Next, set for every $u, v \in V$ and $j \in L$, $c_j(u, v) = \min\{\varphi(u, j), \varphi(v, j)\}$, and $d(u, v) = 1 - \sum_{j \in L} c_j(u, v)$. Thus, if $f(u) = f(v)$, then $c_j(u, v) = 0$ for every $j \neq f(u)$ and $c_{f(u)}(u, v) = 1$. Otherwise, $f(u) \neq f(v)$ and $c_j(u, v) = 0$ for every $j \in L$. The assignments can be viewed in Figure 2.

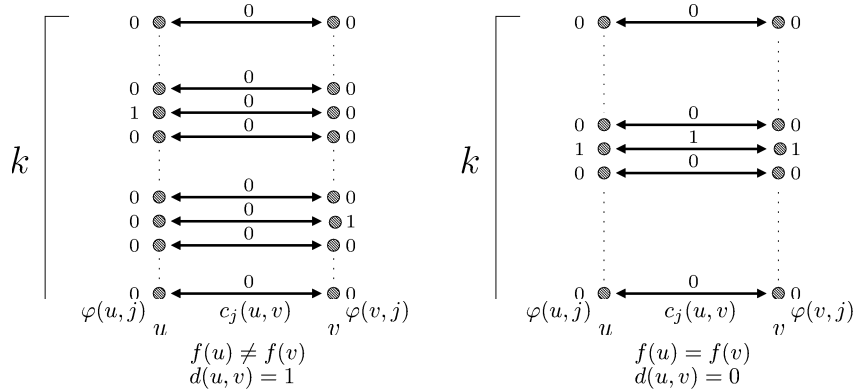


Figure 2: Variable Assignments.

Clearly, this solution satisfies constraints (3) (closeness constraints), (4) (variation distance constraints), and also constraints (1) (probability measure constraints). Since the labeling f does not assign more than ℓ vertices to each label, constraints (2) (capacity constraints) are also satisfied. Note that $d(u, v) = 0$ if $f(u) = f(v)$, and $d(u, v) = 1$ otherwise. Let us now focus on constraints (5) (spreading constraints). Let $S \subseteq V$ and $u \in S$. Since f assigns at most ℓ vertices to each label $j \in L$, in particular for label $f(u)$ we get that: $|\{x \in S \mid f(u) \neq f(x)\}| \geq |S| - \ell$. For every $v \in \{x \in S \mid f(u) \neq f(x)\}$, $d(u, v) = 1$, hence the solution satisfies constraints (5).

Let us now focus on constraints (6) (closeness Δ -inequality constraints). If v and w are assigned the same label ($f(v) = f(w)$), then $d(v, w) = 0$ (or equivalently $\sum_{j \in L} c_j(v, w) = 1$) and $c_j(u, v) = c_j(u, w)$, $\forall j \in L$. Thus, the solution satisfies the constraint in this case. Otherwise, v and w are assigned different labels ($f(v) \neq f(w)$), therefore $d(v, w) = 1$ (or equivalently $\sum_{j \in L} c_j(v, w) = 0$). If u is assigned to the same label as v or w ($f(u) \in \{f(v), f(w)\}$), the left hand side of the constraint equals 1 and the solution

satisfies the constraint. If u is not assigned the same label as v and w ($f(u) \notin \{f(v), f(w)\}$), the left hand side of the constraint equals 0 and the solution satisfies the constraint. Thus, we conclude that constraints (6) hold, and hence the solution is feasible.

We now examine the value of the solution. Clearly, for each vertex $v \in V$,

$$\sum_{j \in L} c(v, j) \varphi(v, j) = c(v, f(v)).$$

Since $d(u, v) = 0$ if $f(u) = f(v)$ and $d(u, v) = 1$ otherwise, we can conclude that $\forall u, v \in V, w(u, v) d(u, v) = w(u, v) d_{\text{unif}}(f(u), f(v))$. Therefore, the value of the objective function is precisely the cost of the labeling, completing the proof. \square

D Proof of Lemma 5

Lemma 5. *An optimal solution for the above program is computable in polynomial time.*

Proof. There is a polynomial number of constraints of types (1), (2), (3), (4), and (6). Thus, we only need to show a separation oracle for the spreading constraints, (5), since there is an exponential number of them. Then we can use the ellipsoid algorithm to compute an optimal solution in polynomial time.

We show how a separation oracle can check in polynomial time if all constraints of this type are satisfied, and otherwise, generate an unsatisfied constraint. For each vertex $u \in V$ and for each possible cardinality s of a subset, we can find the $s - 1$ closest vertices to u by sorting all vertices v (except for $u, v \neq u$) with respect to $d(u, v)$. Therefore, we find a set S of size s that contains u and the sum of distances between u and all other vertices in S is minimal, in comparison to all other possible sets of size s that contain u . Hence, if the set S which the separation oracle finds does not satisfy the spreading constraint, we can return this constraint. Otherwise, since S contains u and the sum of distances between u and all other vertices in S is minimal, in comparison to all other possible sets of size s that contain u , we can conclude that all constraints for the current u and s hold. Thus, we have a polynomial time separation oracle. \square

E Proof of Lemma 6

Lemma 6. *All vertices are labeled by ApproxLabel.*

Proof. For every $v \in V, d(v, v) = 0$. Hence, $c_j(v, v) = \varphi(v, j)$ for every $j \in L$, due to constraints (3) (closeness constraints), (4) (variation distance constraints), and (1) (probability measure constraints). Thus, $u_i \in \{x \mid d(u_i, x) \leq R\} \cap \{x \mid c_{f^*(u_i)}(u_i, x) \geq \alpha\}$, for any possible value of R, i, α , and $f^*(u_i)$. Hence, every u_i is assigned the latest to a label in iteration i . \square

F Proof of Lemma 7

Lemma 7 *For every iteration $1 \leq i \leq n$ and any pair of vertices $u, v \in V$:*

- (a) $\Pr [c_{f^*(u_i)}(u_i, u) \geq \alpha] = 1 - d(u_i, u)$.
- (b) $\Pr [|\{u, v\} \cap \{x \mid c_{f^*(u_i)}(u_i, x) \geq \alpha\}| = 1] \leq d(u, v)$.

Proof. To prove (a), according to Property (7), constraints (3) (closeness constraints), (4) (variation distance constraints), and the fact that $\alpha \sim \text{Unif}[0, \varphi(u_i, f^*(u_i))]$, we obtain that:

$$\begin{aligned} \mathbb{P}\text{r} [c_{f^*(u_i)}(u_i, u) \geq \alpha] &= \sum_{j \in L} \mathbb{P}\text{r} [c_{f^*(u_i)}(u_i, u) \geq \alpha \mid f^*(u_i) = j] \cdot \mathbb{P}\text{r}[f^*(u_i) = j] \\ &= \sum_{j \in L} \frac{c_j(u_i, u)}{\varphi(u_i, j)} \varphi(u_i, j) = 1 - d(u_i, u) \end{aligned}$$

To prove (b), according to Property (7), constraints (3) (closeness constraints), (4) (variation distance), constraints (6) (closeness Δ -inequality constraints), and the fact that $\alpha \sim \text{Unif}[0, \varphi(u_i, f^*(u_i))]$, we obtain that:

$$\begin{aligned} \mathbb{P}\text{r} [|\{u, v\} \cap \{x \mid c_{f^*(u_i)}(u_i, x) \geq \alpha\}| = 1] &= \\ \sum_{j \in L} \mathbb{P}\text{r} [|\{u, v\} \cap \{x \mid c_{f^*(u_i)}(u_i, x) \geq \alpha\}| = 1 \mid f^*(u_i) = j] \cdot \mathbb{P}\text{r}[f^*(u_i) = j] &= \\ \sum_{j \in L} \frac{|c_j(u_i, u) - c_j(u_i, v)|}{\varphi(u_i, j)} \varphi(u_i, j) \leq 1 - \sum_{j \in L} c_j(u, v) = d(u, v) \end{aligned}$$

□

G Proof of Lemma 9

Lemma 9. For every iteration $1 \leq i \leq n$, every label $j \in L$, and every vertex $v \in V$:

$$\mathbb{P}\text{r} [(c_{f^*(u_i)}(u_i, v) \geq \alpha) \wedge (f^*(u_i) = j)] = c_j(u_i, v).$$

Proof. It follows from Property (7), constraint (3) (closeness constraints), and the additional fact that $\alpha \sim \text{Unif}[0, \varphi(u_i, f^*(u_i))]$, that:

$$\begin{aligned} \mathbb{P}\text{r} [(c_{f^*(u_i)}(u_i, v) \geq \alpha) \wedge (f^*(u_i) = j)] &= \mathbb{P}\text{r} [c_{f^*(u_i)}(u_i, v) \geq \alpha \mid f^*(u_i) = j] \cdot \mathbb{P}\text{r}[f^*(u_i) = j] \\ &= \frac{c_j(u_i, v)}{\varphi(u_i, j)} \cdot \varphi(u_i, j) = c_j(u_i, v). \end{aligned}$$

□

H Proof of Theorem 11

Theorem 11. The expected cost of a solution of **ApproxLabel** is at most:

$$\left(\frac{1 + \varepsilon}{\varepsilon} \ln(n) + 1 + \varepsilon \right) \left(\frac{2n}{n-1} \right) \cdot OPT.$$

Proof. Define the following random variables:

- $Y_{u,v}$ - Indicator for the event that vertices u and v are assigned to different labels by **ApproxLabel**.

- $Y_{v,j}$ - Indicator for the event that v is assigned to label j .

By Lemma 8, $\mathbb{E}[Y_{u,v}] \leq \mathbb{P}[A(u,v)|C_0(u,v)] \leq (\tau + 1 + \varepsilon) \left(\frac{2n}{n-1}\right) d(u,v)$, and by Lemma 10, $\mathbb{E}[Y_{v,j}] = \mathbb{P}[A(v,j)|C_0(v)] \leq (1 + \varepsilon) \varphi(v,j)$. Hence, the expected cost of the solution found by **ApproxLabel** is:

$$\begin{aligned} & \mathbb{E} \left[\sum_{v \in V} \sum_{j \in L} c(v,j) Y_{v,j} \right] + \mathbb{E} \left[\sum_{u,v \in V} w(u,v) Y_{u,v} \right] \leq \\ & \left(\frac{1 + \varepsilon}{\varepsilon} \ln n + 1 + \varepsilon \right) \left(\frac{2n}{n-1} \right) \cdot \left(\sum_{v \in V} \sum_{j \in L} c(v,j) \varphi(v,j) + \sum_{u,v \in V} w(u,v) d(u,v) \right) \end{aligned}$$

The theorem now follows from Lemma 4. \square

I Proof of Lemma 12

Lemma 12. *Assume (V, d) is a semi-metric that satisfies the spreading constraints (5). Then, for every $S \subseteq V$ such that $|S| > \gamma \ell$ ($\gamma > 1$) and every $u \in S$, there exists a vertex $v \in S$ such that $d(u,v) > 1 - \frac{1}{\gamma}$.*

Proof. Let $S \subseteq V$ such that $|S| > \gamma \ell$ ($\gamma > 1$) and $u \in S$. According to constraints (5), $\sum_{v \in S} d(u,v) \geq |S| - \ell$. Therefore, there exists $v \in S$ such that $d(u,v) \geq 1 - \frac{\ell}{|S|}$. Since $|S| > \gamma \ell$, we obtain that $d(u,v) > 1 - \frac{1}{\gamma}$. \square

J Proof of Lemma 14

Lemma 14. *For every $v \in V$, and every $j \in L$, $Y_{v,j}$ is independent of $\left\{ Y_{u,j} \mid d(u,v) > 2 \frac{\varepsilon}{1+\varepsilon} \right\}$.*

Proof. Let us denote $X_j(v) = \left\{ Y_{u,j} \mid d(u,v) > 2 \frac{\varepsilon}{1+\varepsilon} \right\}$. Assume to the contrary that $Y_{v,j}$ is not independent of $X_j(v)$. Thus, there exists a vertex $w \in V$ such that in the iteration in which w is the root, there is a positive probability that both v and some of the vertices in $X_j(v)$ are assigned to w 's root label (notice that it might be the case that $w = v$ or $w \in X_j(v)$). This is true since $\{f^*(u_i)\}_{1 \leq i \leq n}$ are independent, and all R 's and α 's chosen by **ApproxLabel** are also independent.

All vertices that are assigned to a label in the iteration in which w is the root are in a sphere centered at w having radius of at most $\frac{\varepsilon}{1+\varepsilon}$. By the triangle inequality for d (Lemma 3), we get that $d(u,v) \leq 2 \frac{\varepsilon}{1+\varepsilon}$ for every two vertices u and v that are assigned to a label in that iteration. Since there is a positive probability that v and some of the vertices in $X_j(v)$ are assigned to a label in the iteration that w is the root, we conclude that there must exist a vertex $x \in X_j(v)$ such that $d(v,x) \leq 2 \frac{\varepsilon}{1+\varepsilon}$. This is a contradiction. \square

K Upper Bound of $(\ell + 1)(1 + \varepsilon)\ell$

In order to achieve this upper bound, we interpret $\{\varphi(u)\}_{u \in V}$ as an $n \times k$ matrix where the row that corresponds to v is $\varphi(v)$. We denote this matrix by Φ , and additionally we denote by an upper index i , the i -th column of a matrix. The procedure creates a new matrix of size $n \times n$, denoted by A , that is double stochastic

(the sum of each row and column is 1). This is done by going over the columns of Φ according to their index, and “aggregating” and “splitting” them, such that each new column has exactly a sum of 1. This process can be done, since the sum of all elements in Φ is n (the sum of each row is 1 and there are n rows). This process consists of stages 1 to 6 in the algorithm.

For example, assume that the first column of Φ has a sum of $\frac{3}{2}$, the second column a sum of $\frac{1}{4}$ and the third column a sum of $\frac{1}{2}$. Then, $A^1 = \frac{2}{3}\Phi^1$, and $A^2 = \frac{1}{3}\Phi^1 + \Phi^2 + \frac{1}{2}\Phi^3$.

After constructing A , we sample a random permutation matrix A^* from a Birkhoff decomposition of A . A Birkhoff decomposition of a double-stochastic matrix A , is a convex finite combination of permutation matrices that equals A . Specifically, $A = \sum_{i=1}^N \lambda_i A_i$ where $\sum_{i=1}^N \lambda_i = 1$, and for every $1 \leq i \leq N$, $\lambda_i \geq 0$ and A_i is a permutation matrix. It is well known that such a decomposition can be constructed in polynomial time. One can view $\{\lambda_i\}_{1 \leq i \leq N}$ as a distribution over the permutation matrices that appear in the decomposition. Hence, when we “sample” from a Birkhoff decomposition, we choose a permutation matrix according to this distribution.

We use the permutation matrix A^* to sample a label for each vertex in the following way. Since A^* is a permutation matrix, for every $u_i \in V$ there is a unique column j' in A^* such that $A^*(u_i, j') = 1$. We choose a random label from all labels that were “aggregated” into column j' according to their relative value in column j' with respect to vertex u_i . This process consists of stages 7 to 12 in the algorithm.

Consider the previous example, where for vertex v , $\varphi(v, 1) = \frac{1}{2}$, $\varphi(v, 2) = \frac{1}{4}$ and $\varphi(v, 3) = \frac{1}{4}$. If $A^*(v, 1) = 1$, then vertex v is assigned a root label of 1 with probability 1. Additionally, if $A^*(v, 2) = 1$, then vertex v is assigned a root label of 1 with probability $\frac{4}{13}$, a root label of 2 with probability $\frac{6}{13}$ and a root label 3 with probability $\frac{3}{13}$ (recall that according to the Birkhoff decomposition, $\mathbb{P}[A^*(v, 1) = 1] = A(v, 1)$ and $\mathbb{P}[A^*(v, 2) = 1] = A(v, 2)$).

Algorithm RootLabel($\{\varphi(u)\}_{u \in V}$)

1. For $i = 0$ to k do:
2. Set $a_i \leftarrow \sum_{j=1}^i \sum_{v \in V} \Phi(v, j)$.
3. For $i = 1$ to n do:
4. Set $i_{\min} \leftarrow \max\{i' | a_{i'} \leq i - 1\}$ and $i_{\max} \leftarrow \min\{i' | a_{i'} \geq i\}$.
5. If $i_{\min} + 1 = i_{\max}$, set $A^i \leftarrow \Phi^{i_{\max}} / (a_{i_{\max}} - a_{i_{\min}})$.
6. If $i_{\min} + 1 \neq i_{\max}$,
 set $A^i \leftarrow \Phi^{i_{\min}+1} \cdot \frac{a_{i_{\min}+1} - (i-1)}{a_{i_{\min}+1} - a_{i_{\min}}} + \sum_{t=i_{\min}+2}^{i_{\max}-1} \Phi^t + \Phi^{i_{\max}} \cdot \frac{i - a_{i_{\max}-1}}{a_{i_{\max}} - a_{i_{\max}-1}}$.
7. Sample a random permutation matrix A^* from a Birkhoff decomposition of A .
8. For every $v \in V$ do:
9. Let j' be the unique label such that $A^*_{v, j'} = 1$.
10. Assume $A^{j'} = \sum_{i=1}^n \alpha_i \Phi^i$ (notice that some α 's might equal 0).
11. Set $f^*(v) \leftarrow j$ with probability $\frac{\alpha_j \Phi(v, j)}{A(v, j)}$.
12. Output f^* .

Clearly Property (7) holds. Additionally, it is easy to see that there are at most $\ell + 1$ vertices assigned to a single label in f^* . This is true since each column Φ^i has a total value of at most ℓ (this is due to constraints (2), the capacity constraints). Thus, each column of Φ can be “spread” on at most $\ell + 1$ columns of A . Therefore, there will always be at most $\ell + 1$ vertices assigned to a single label in f^* .

In a single iteration of **ApproxLabel**, all vertices that are assigned to a label in that iteration have a

distance of at most $\frac{\varepsilon}{1+\varepsilon}$ from that iteration's root. According to Corollary 13, there are at most $(1 + \varepsilon)\ell$ such vertices. Therefore, we obtain an upper bound of $(\ell + 1)(1 + \varepsilon)\ell$ vertices that are assigned to the same label.