Capacitated Metric Labeling

Matthew Andrews * Mohammad Taghi Hajiaghayi † Howard Karloff ‡
Ankur Moitra §

Abstract

We introduce Capacitated Metric Labeling. As in Metric Labeling, we are given a weighted graph $G = (V, E)$, a label set L, a semimetric d_L on this label set, and an assignment cost function $\phi : V \times L \to \Re^+$. The goal in Metric Labeling is to find an assignment $f : V \to L$ that minimizes a particular two-cost function. Here we add the additional restriction that each label t_i receive at most l_i nodes, and we refer to this problem as Capacitated Metric Labeling. Allowing the problem to specify capacities on each label allows the problem to more faithfully represent the classification problems that Metric Labeling is intended to model.

Our main positive result is a polynomial-time, $O(\log |V|)$-approximation algorithm when the number of labels is fixed, which is the most natural parameter range for classification problems. We also prove that it is impossible to approximate the value of an instance of Capacitated Metric Labeling to within any finite factor, if $P \neq NP$.

Yet this does not address the more interesting question of how hard Capacitated Metric Labeling is to approximate when we are allowed to violate capacities. To study this question, we introduce the notion of the "congestion" of an instance of Capacitated Metric Labeling. We prove that (under certain complexity assumptions) there is no polynomial-time approximation algorithm that can approximate the congestion to within $O((\log |L|)^{1/2-\epsilon})$ (for any $\epsilon > 0$) and this implies as a corollary that any polynomial-time approximation algorithm that achieves a finite approximation ratio must multiplicatively violate the label capacities by $\Omega((\log |L|)^{1/2-\epsilon})$. We also give a $O(\log |L|)$-approximation algorithm for congestion.

1 Introduction

We introduce Capacitated Metric Labeling. As in Metric Labeling (introduced by Kleinberg and Tardos in [21]), we are given a weighted graph $G = (V, E)$ (with a weight function $w : E \to \Re^+$), a label set L, a semimetric d_L on this label set, and an assignment cost function $\phi : V \times L \to \Re^+$. The goal in Metric Labeling is to find an assignment $f : V \to L$ that minimizes a two-cost function:

$$\sum_u \phi(u, f(u)) + \sum_{\{u,v\} \in E, u<v} w(u, v) d_L(f(u), f(v)).$$

Here we consider the same optimization problem, but subject to the additional restriction that each label t_i is assigned a capacity l_i and we restrict the assignment function f to assign at most l_i nodes to label t_i. We refer to this problem as Capacitated Metric Labeling. Throughout this paper, we will use $n = |V|$ and $k = |L|$.

Metric Labeling was introduced as a model to study a broad range of classification problems that arise in computer vision and related fields. Yet allowing the problem to specify a capacity on each label actually allows the problem to more faithfully represent the classification problems that Metric Labeling was intended to model. In particular, placing capacities on labels restricts the space of near-optimal solutions in such a way that these solutions are much more consistent with abstract notions of what a "good" classification should look like. Capacitated Metric Labeling can also be seen as natural from a theoretical standpoint, and generalizes many well-studied problems including Metric Labeling (and all the problems that Metric Labeling generalizes) and balanced partitioning problems such as Balanced Metric Labeling, introduced in [24].

Our main positive result is a polynomial-time, $O(\log n)$-approximation algorithm when the number of labels is fixed. This extends Räcke's results on Minimum Bisection to a constant number of labels, to a more general cost function, and to arbitrary capacities for each label – but is based on the same paradigm as in [26] of rounding the graph into a decomposition tree, and then using dynamic programming to solve the exact problem on a tree. A constant number of labels is the most natural parameter range for classification problems, and in this regime we give a substantial improvement over results in [24] which work only for a uniform metric space (on the labels) and uniform label capacities; in addition, their algorithm needs to cheat on label capacities by a $\Omega(\log k)$ factor, when compared to the optimal solution. Our approach to this result is particularly simple, and we give a method of combining the

*andrews@research.bell-labs.com. Bell Labs.
†hajiagha@mit.edu. ATT Labs—Research.
‡howard@research.att.com. ATT Labs—Research.
§moitra@mit.edu. MIT. This research was supported in part by a Fannie and John Hertz Foundation Fellowship. Part of this work was done while the author was visiting ATT Labs—Research and Bell Labs, and supported in part by DIMACS.

recent results of Räcke on hierarchical decomposition trees [26] with earthmover relaxations. Using Räcke's results to approximate the original problem by a corresponding problem on a tree, we can then interpret any feasible solution to an appropriately chosen earthmover relaxation as giving us conditional probabilities on every edge in the tree. Because the graph actually is a tree, there will be a rounding procedure that is consistent with these conditional probabilities, and the expected cost will be equal to the value of the LP.

How hard is Capacitated Metric Labeling to approximate? In order to study the inapproximability of Capacitated Metric Labeling, we introduce the notion of the "congestion": Given an instance of Capacitated Metric Labeling, the *congestion* is the minimum value $C \geq 1$ so that if we were to scale up the label capacities by a factor of C, there would be a valid assignment that has zero cost. We explain the relationship between congestion and hardness in detail in Section 1.2, but for now we note that congestion is a framework in which to study the question: How much must a polynomial-time approximation algorithm inflate the capacities in order to obtain *any* finite approximation ratio for Capacitated Metric Labeling? We prove that (under certain complexity assumptions), that there is no polynomial-time approximation algorithm that can approximate the congestion to within $O((\log k)^{1/2-\epsilon})$ for any $\epsilon > 0$, via a reduction from a wavelength assignment problem [4]. And this yields an answer to the above question, as follows: for any $\epsilon > 0$, no polynomial-time approximation algorithm that achieves a finite approximation ratio can always violate the label capacities by $O((\log k)^{1/2-\epsilon})$. We will refer to this as a $(\infty, \Omega((\log k)^{1/2-\epsilon}))$-bicriteria hardness result for Capacitated Metric Labeling.

Our inapproximability results for the congestion version of Capacitated Metric Labeling are not far from tight: we demonstrate a $O(\log k)$-approximation algorithm for determining the congestion of an instance of Capacitated Metric Labeling. This is important because it means that we can approximately determine if there is a zero-cost solution or not, and such a step may be a building block in giving a general bicriteria approximation algorithm that achieves a polylogarithmic approximation ratio in the cost, and which violates capacities by only a polylogarithmic factor. To achieve this result, we introduce a constraint in our LP that bounds the change in the expected number of nodes assigned to any label, after any choice of the rounding algorithm. Our technique can be regarded as an analogue of the randomized rounding techniques for flows. And so, just as we cannot hope for better than $(1, O(\frac{\log n}{\log \log n}))$-bicriteria hardness for Undirected Edge Disjoint Paths, we also cannot hope for better than $(\infty, \Omega(\log k))$-bicriteria hardness for Capacitated Metric Labeling.

1.1 Metric Labeling and Image Segmentation

Introduced by Kleinberg and Tardos [21], Metric Labeling models a broad range of classification problems arising in computer vision and related fields. Typically, in a classification problem, one is given a set of points and a set of labels and the goal is to find some labeling of the points that optimizes some notion of quality. This notion of quality is usually defined based on some observed data about the problem. In particular, Metric Labeling captures classification problems in which one is given information about the pairwise relations between the underlying points.

Intuitively, the assignment costs $\phi(u, i)$ capture how likely the label t_i is for the node u, and the weights $w(u, v)$ capture the strength of the pairwise relation between u and v. Pairs of nodes with a stronger pairwise strength are more expensive to separate among different labels.

Naor and Schwartz [24] introduced Balanced Metric Labeling (a special case of Capacitated Metric Labeling) in which each label is given the same capacity l, and additionally the semimetric d_L on the labels is uniform. Reference [24] gives a bicriteria approximation algorithm that achieved a $O(\log n)$ approximation in the cost function but only while violating label capacities by a factor which is $O(\min\{l, \log k\})$. Our algorithm for Capacitated Metric Labeling, by contrast, does not violate capacities when k is $O(1)$ (and it works for arbitrary metrics). Again, the motivation in considering this problem was applications to image segmentation problems.

A generalization of both of these problems, Capacitated Metric Labeling is much more expressive than both Metric Labeling and Balanced Metric Labeling in a number of important ways that help the problem better capture classification questions in computer vision, and related fields. Consider, as an example, an image segmentation problem in which one is given a raster (or array) of pixel values and one wants to segment the image into a number of categories such as

$$\Big\{\textbf{sky, ground, people, buildings,...}\Big\}.$$

Then pixels which are blue, for example, will have large assignment cost when mapped to any label other than **sky**. But consider the effect of pairwise relationships on the image segmentation question. If the picture is mostly composed of sky, then perhaps the pairwise costs of separating pixels into sky and ground will overwhelm the cost of mistakenly labeling ground pixels as sky.

Here, a Metric Labeling approximation algorithm may return a trivial and uninformative solution in which the entire image is labeled as sky. This is a common pitfall in image segmentation problems.

What about Balanced Metric Labeling? In addition to the obvious deficiency that the approximation algorithm of Naor and Schwartz [24] requires that the label metric be uniform, this scheme also requires that the capacities on the labels be uniform too. But if we know *a priori* that the image should be mostly sky, then we would benefit greatly from being able to set upper bounds on the number of pixels labeled as $\{sky, ground, people, buildings, ...\}$ independently. Yet Capacitated Metric Labeling can in principle overcome both of these shortcomings.

Theorem 1.1. *There is a $O(\log n)$-approximation algorithm for* Capacitated Metric Labeling *that runs in time $(n+1)^{3k} poly(n, k)$ (and does not violate label capacities), where $k = |L|$.*

For general k, under certain complexity assumptions, there is no polynomial-time algorithm that achieves *any* finite approximation ratio, and we will discuss this result in more detail in Section 1.2.

As we noted earlier, our approximation algorithm is based on combining the recent results of Räcke on hierarchical decomposition trees [26] and earthmover relaxations. Räcke's results can be seen as a dual to the well-known problem of embedding a metric space into a distribution on dominating trees [7], [15]. These results on embeddings into distributions over dominating trees are the building blocks of previous approximation algorithms for both Metric Labeling [21] and Balanced Metric Labeling [24]. Yet by using these tools in a dual form (i.e., using Räcke's results) the description of the approximation algorithm and the proofs all become much simpler. We need no tools for obtaining concentration for weakly dependent random variables; instead all we need is elementary conditional probability. In this way, our results are also a simplification of previous approximation algorithms in this line of work.

We also apply our techniques to Balanced Metric Labeling, and for this problem (where l is the uniform capacity on the labels), we obtain a $O(\log l)$-approximation algorithm that violates capacities multiplicatively by $O(\log k)$. We achieve a slightly better approximation ratio, but at the price of a worse violation of capacities (in some cases) compared to the result in [24], which gives a $O(\log n)$-approximation ratio but violates capacities multiplicatively by $O(\min\{\log k, l\})$.

Theorem 1.2. *There is a polynomial-time, $O(\log l)$-approximation algorithm for* Balanced Metric Labeling *that multiplicatively violates capacities by $O(\log k)$.*

1.2 Congestion and Bicriteria Hardness How hard is Capacitated Metric Labeling to approximate? Assuming $P \neq NP$, there is no polynomial time constant factor approximation algorithm for $(k, 1)$ Balanced Graph Partitioning [1], and $(k, 1)$ Balanced Graph Partitioning[1] is a special case of Capacitated Metric Labeling. So assuming $P \neq NP$, it is impossible to approximate the value of an instance of Capacitated Metric Labeling to within any bounded factor if one is not allowed to violate capacities. The reduction in [1] is from 3-Partition, and in fact we give an improved reduction from 3-Partition to the more general Capacitated Metric Labeling and this reduction implies that it is NP-hard to approximate the value of an instance of Capacitated Metric Labeling to within even an $n^{1-\epsilon}$ factor.

Yet, this does not address the more interesting question of how hard Capacitated Metric Labeling is to approximate when one is are allowed to violate capacities. To study this question, we introduce the notion of congestion.

Defintion 1.1. *Given an instance of* Capacitated Metric Labeling*, the congestion is the minimum value $C \geq 1$ so that if one were to scale up the label capacities by a factor of C, there would be a valid assignment that has zero cost.*

The congestion can only take nk possible values because at least one scaled label capacity Cl_i must be an integer at most n. So if there were a polynomial-time approximation algorithm for Capacitated Metric Labeling that achieved *any* finite approximation ratio, then whenever the Capacitated Metric Labeling instance has a zero-cost solution (and only then), the approximation algorithm would need to return a zero-cost solution, too. Hence, there would be a polynomial-time algorithm for determining the congestion of an instance of Capacitated Metric Labeling.

Yet we prove that (under certain complexity assumptions), for all $\epsilon > 0$, there is no polynomial-time approximation algorithm that can approximate the congestion to within a factor which is $O((\log k)^{1/2-\epsilon})$.

Theorem 1.3. *If $NP \not\subseteq ZPTIME(n^{polylog\ n})$, then for all $\epsilon > 0$, there is no polynomial-time approximation algorithm for determining the congestion of a* Capacitated Metric Labeling *instance to within a factor*

[1]The goal of (k, ν) Balanced Graph Partitioning is to partition the graph into k parts, each of size at most $\nu\frac{n}{k}$ so as to minimize the total capacity crossing between different parts.

which is $O((\log k)^{1/2-\epsilon})$. This holds even if the distances in the label semimetric are restricted to be zero or one.

This result rules out the possibility of a polynomial-time approximation algorithm that achieves any finite approximation ratio. In fact, this result is even strong enough to obtain bicriteria hardness results: Suppose we allowed our approximation algorithm to violate the capacities of each label by a factor of C, but we still compared the cost to the cost of the optimal assignment which is not allowed to violate label capacities. If such a polynomial-time bicriteria approximation algorithm existed (which achieved *any* finite approximation ratio), it would imply that there is a polynomial-time algorithm to distinguish between the case in which there is a zero-cost assignment, and the case in which there is no zero-cost assignment even if the label capacities are scaled up by a factor of C.

Corollary 1.1. *If $NP \not\subseteq ZPTIME(n^{polylog\ n})$, then for all $\epsilon > 0$, there is no polynomial-time approximation algorithm for* Capacitated Metric Labeling *that achieves a finite approximation ratio in the cost and violates the capacities of labels multiplicatively by $O((\log k)^{1/2-\epsilon})$.*

So the notion of congestion that we introduce here is a natural framework in which to study the question, how much must a polynomial-time approximation algorithm "cheat" (i.e., inflate the capacities) in order to obtain any finite approximation ratio for Capacitated Metric Labeling? Our results imply that for any $\epsilon > 0$, any polynomial-time approximation algorithm that achieves a finite approximation ratio must multiplicatively violate the label capacities by $\Omega((\log k)^{1/2-\epsilon})$ for some inputs.

Defintion 1.2. *We will say that a problem is $(f(n), g(k))$-bicriteria hard (under some complexity assumption) if there is no polynomial-time algorithm that achieves an $f(n)$-approximation in the cost ($n = |V|$), and cheats on capacities by a factor of $g(k)$ ($k = |L|$).*

In this terminology, Capacitated Metric Labeling is $(\infty, \Omega((\log k)^{1/2-\epsilon}))$-bicriteria hard (assuming $NP \not\subseteq ZPTIME(n^{polylog\ n})$). Our hardness results are inspired by those obtained in [3], [2] and [10]. The first inapproximability results for Edge Disjoint Paths in undirected graphs were obtained in [5]. This result was subsequently strengthened in [3] (see also [2]) to give bicriteria hardness: Suppose our approximation algorithm for Edge Disjoint Paths is allowed to route at most C paths through an edge, but we compare our solution to the optimal solution, which routes at most one path through each edge. Then [3] proves that (under certain complexity assumptions) there is no polynomial-time approximation algorithm that achieves an approximation ratio that is $O(\log^{\frac{1-\epsilon}{C+1}} n)$, for all $\epsilon > 0$. Using our terminology, Undirected Edge Disjoint Paths is $(O(\log^{\frac{1-\epsilon}{C+1}} n), C)$-bicriteria hard (assuming $NP \not\subseteq ZPTIME(n^{polylog\ n})$) and this holds for any $C = C(k)$ which is $o\left(\frac{\log \log k}{\log \log \log k}\right)$.

Our inapproximability results for the congestion version of Capacitated Metric Labeling are almost tight: we demonstrate a $O(\log |L|)$-approximation algorithm for determining the congestion of an instance of Capacitated Metric Labeling.

Theorem 1.4. *There is a $O(\log k)$-approximation algorithm for determining the congestion of an instance of* Capacitated Metric Labeling.

As we noted earlier, our approximation algorithm for this problem is based on preemptively enforcing that each possible choice in our randomized rounding algorithm cannot increase the expected number of nodes assigned to any label by too much. This gives us a way to interpret the rounding procedure as a martingale in which the random variable is the conditional expectation of the number of nodes assigned to any label (conditioned on the incremental choices of the rounding algorithm), and we are able to enforce that each step does not increase this random variable too much. In this way, we obtain tight concentration, and can obtain an approximation algorithm for the congestion.

This can be regarded as an analogue of randomized rounding for flows. Just as randomized rounding rules out $(1, \omega(\frac{\log n}{\log \log n}))$-bicriteria hardness for Undirected Edge Disjoint Paths, we also cannot hope for $(\infty, \omega(\log k))$-bicriteria hardness for Capacitated Metric Labeling (by Theorem 1.4).

Can this rounding procedure be used to give a bicriteria approximation algorithm that violates capacities by a $O(\log k)$ multiplicative factor, and achieves a polylogarithmic approximation ratio in the cost? We don't know, but we demonstrate that for a natural generalization of the relaxations used in [9] and [24] to nonuniform spreading metrics, the natural linear program has an integrality ratio of $\Omega(n^{1/5})$.

2 Definitions

In Metric Labeling, we are given an undirected, capacitated graph $G = (V, E)$. Let $w : E \rightarrow \Re^+$ be the weight function. We let k denote the number of labels. We are given a semimetric d_L on the label set $L = \{t_1, t_2, ..., t_k\}$. We let $\phi(u, t_i)$ denote the cost of assigning node u to label t_i. The goal of Metric La-

BELING is to find a function $f : V \to L$ that minimizes $\sum_u \phi(u, f(u)) + \sum_{\{u,v\} \in E, u<v} w(u, v) d_L(f(u), f(v))$. What distinguishes CAPACITATED METRIC LABELING from METRIC LABELING is that in the former we are also given a capacity l_i for label t_i. (Note that $l_i = l$ for all i, together with a uniform metric, is the special case that corresponds to BALANCED METRIC LABELING as defined in [24].) Then additionally (in CAPACITATED METRIC LABELING) we require that f satisfy (for all i): $|\{u | f(u) = t_i\}| \leq l_i$.

Throughout this paper, we will use $n = |V|$ and $k = |L|$.

2.1 Hierarchical Decompositions

Räcke recently gave an algorithm to construct asymptotically optimal oblivious routing schemes in undirected graphs [26]. These oblivious routing schemes are $O(\log n)$-competitive. In fact, Räcke proved a stronger statement, and constructed these oblivious routing schemes by constructing optimal hierarchical decompositions for approximating the cuts of a graph.

Räcke constructed hierarchical decompositions via decomposition trees. In order to describe these formally, we first give some preliminary definitions. We let $G = (V, E)$ be an undirected, weighted graph on $|V| = n$ nodes and $|E| = m$ edges. Additionally, we let $w : E \to \Re^+$ be the weight function of this graph.

DEFINTION 2.1. *Given a subset $U \subseteq V$, we define $w_G(U) = \sum_{\{u,v\} \in E \text{ s.t. } |\{u,v\} \cap U| = 1, u<v} w(u, v)$.*

DEFINTION 2.2. *A decomposition tree T for G is a weighted tree on V (not necessarily a subgraph of G) along with a function f_T that maps edges $e = \{u, v\}$ in T to paths $f_T(e)$ connecting u and v in G. Additionally, the weight of an edge in T is chosen as follows. Let $(U_e, V - U_e)$ be the partition of V that results from deleting e from T. Then we set $w_T(e) = w_G(U_e)$.*

Where convenient, we will also write $f_T(u, v)$ instead of $f_T(e)$, if the edge e connects u and v. So we can regard a decomposition tree as exactly representing some of the cuts in G. Additionally, the function f_T will allow us to define the notion of the load on an edge e in G associated with a decomposition tree. This notion of load is related to how well the decomposition tree approximates all cuts in G.

DEFINTION 2.3. *Given a decomposition tree T for G, $e \in E(G)$, and the mapping function f_T, $load_T(e) = \sum_{\{u,v\} \in E_T \text{ s.t. } f_T(u,v) \ni e} w_T(u, v)$.*

Then we can state the main theorem in [26] as

THEOREM 2.1. *[Räcke] [26] There is a polynomial-time algorithm which, for any undirected, weighted graph G, constructs a distribution μ on decomposition trees $T_1, ..., T_r$ such that for any edge $e \in G$, $E_{T \leftarrow \mu}[load_T(e)]$ is $O(\log n) w(e)$, and r is $O(mn \log n)$ where n is the number of nodes and m is the number of edges in G.*

We will use this result extensively to get bicriteria approximation algorithms for CAPACITATED METRIC LABELING.

3 Reductions via Hierarchical Decompositions

Here, we use Räcke's results [26] on hierarchical decompositions to reduce general CAPACITATED METRIC LABELING to an identical problem on trees, at the expense of a $O(\log n)$ factor in the approximation ratio. Suppose we fix the label semimetric d_L, the assignment function ϕ, and the label capacities $< l_1, l_2, ..., l_k >$.

DEFINTION 3.1. *Let $cost(G, L, d_L, \phi, < l_1, l_2, ..., l_k >, f)$ denote the cost of the solution $f : V \to L$ to* CAPACITATED METRIC LABELING *on G, L, d_L, ϕ, and $< l_1, l_2, ..., l_k >$ (provided that f satisfies the label capacity constraints). Let $OPT(G, L, d_L, \phi, < l_1, l_2, ..., l_k >)$ denote the cost of an optimal solution.*

DEFINTION 3.2. *Consider a family of functions $\gamma(u, \cdot) : L \to \Re^+$ for each $u \in V$, and $\gamma(u, \cdot, v, \cdot) : L \times L \to \Re^+$ for each $u, v \in V$. We call this family of functions γ a* fractional assignment *if*

$$\gamma(u, t_i, v, t_j) = \gamma(v, t_j, u, t_i) \quad \forall u, v \in V, u \neq v, i, j \in [l]$$

$$\sum_{j \in [l]} \gamma(u, t_i, v, t_j) = \gamma(u, t_i) \quad \forall u, v \in V, u \neq v, i \in [l]$$

$$\sum_{i \in [l]} \gamma(u, t_i) = 1 \quad \forall u \in V$$

$$\gamma(u, t_i, v, t_j) \geq 0 \quad \forall u, v \in V, u \neq v, i, j \in [l]$$

$$\gamma(u, t_i) \geq 0 \quad \forall u \in V, i \in [l].$$

Additionally, we require that $\delta_\gamma : V \times V \to \Re^+$ where $\delta_\gamma(u, v) = \sum_{i,j} \gamma(u, t_i, v, t_j) d_L(t_i, t_j)$ be a semimetric.

When it is clear from context, we will omit L, d_L, ϕ, and $< l_1, l_2, ..., l_k >$ from the notation used to describe both the cost of a valid solution and the optimal solution. So we will often write $cost(G, f)$ as shorthand for $cost(G, L, d_L, \phi, < l_1, l_2, ..., l_k >, f)$ and similarly $OPT(G)$ for $OPT(G, L, d_L, \phi, < l_1, l_2, ..., l_k >)$.

We will also be interested in fractional solutions γ. For each $u \in V$, $\gamma(u, \cdot)$ will be a probability distribution indexed by the labels, that is, $\gamma(u, t_i) \geq 0$ for all u and i and $\sum_i \gamma(u, t_i) = 1$ and for all u. We would like to extend the notion of the cost of a solution to a fractional solution as well. Then we also need to have an estimate for the probability that the endpoints of an edge are mapped to different labels. For example, the edge $\{u, v\}$ could have $\gamma(u, \cdot) = \gamma(v, \cdot)$, meaning that for each label t_i, both u and v are assigned the same probability of being mapped to t_i. Yet, we could interpret there probability distributions $\gamma(u, \cdot), \gamma(v, \cdot)$ many different ways. We could sample a label a from $\gamma(u, \cdot)$ and a label b from $\gamma(v, \cdot)$ independently, and assign u to a and v to b. Then for some distributions $\gamma(u, \cdot), \gamma(v, \cdot)$, this procedure would map u and v to different labels very often. But we could instead sample a single label a from $\gamma(u, \cdot) = \gamma(v, \cdot)$ and map both u and v to a, and in this procedure we would never map u and v to different labels. So in general, these marginal distributions $\gamma(u, \cdot), \gamma(v, \cdot)$ can have very different realizations as a joint distribution, and this joint distribution is really what determines the cost of a solution – in the example above, we give two different realizations of the same marginal distributions. One of these realizations almost always incurs some cost because u and v are almost always separated, and the other realization never incurs cost because u and v are never separated.

So in order to charge a cost to a fractional solution, we will require that this fractional solution contain information about pairwise probabilities that an edge is split. In particular, we will be given a function $\gamma(u, t_i, v, t_j)$ which describes the probability that u is assigned to label t_i and v is assigned to label t_j. We will additionally need these to be consistent with each $\gamma(u, \cdot)$ distribution. We formalize this notion in Definition 3.2.

Note that a fractional assignment is certainly not necessarily in the convex hull of assignments – just as a fractional solution to an earthmover relaxation for the 0-extension problem [6], [9] is not necessarily a convex combination of 0-extensions. Additionally, we can associate any assignment $f : V \to L$ with a fractional assignment by setting $\gamma_f(u, t_i) = 1$ if $f(u) = t_i$ and 0 otherwise, and $\gamma_f(u, t_i, v, t_j) = 1$ if $f(u) = t_i$ and $f(v) = t_j$, and 0 otherwise. This function γ_f also satisfies the semimetric condition because $\sum_{i,j} \gamma(u, t_i, v, t_j) d_L(t_i, t_j) = d_L(f(u), f(v))$. Then we can abuse notation and refer to an assignment f as a fractional assignment γ_f as well.

We extend the notion of cost to fractional assignments as follows:

Defintion 3.3. *The cost of a fractional assignment γ on a graph G is $cost(G, \gamma) = \sum_{u,i} \phi(u, t_i)\gamma(u, t_i) + \sum_{\{u,v\} \in E} w(u, v) \sum_{i,j} \gamma(u, t_i, v, t_j) d_L(t_i, t_j)$.*

Also we will break this cost up into a Type A cost, and a Type B cost:

Defintion 3.4.

$$\begin{aligned} costA(G, \gamma) &= \sum_{u,i} \phi(u, t_i)\gamma(u, t_i), \textit{ and} \\ costB(G, \gamma) &= cost(G, \gamma) - costA(G, \gamma). \end{aligned}$$

In the case that f is an assignment ($f : V \to L$), if γ_f is the corresponding fractional assignment, then $cost(G, f) = cost(G, \gamma_f)$. So the cost of a fractional assignment generalizes the notion of cost for assignments $f : V \to L$. We will use this more general notion of cost to prove that not only does the cost of an optimal assignment not increase by too much when rounding the graph G to an appropriately chosen decomposition tree T, but even the cost of any particular fractional assignment does not increase by too much.

Lemma 3.1. *[Fractional Decomposition Lemma] Given any fractional assignment γ and any decomposition tree T, $cost(T, \gamma) \geq cost(G, \gamma)$. Also if we let μ be a distribution on decomposition trees with, for all $e \in G$, $E_{T \leftarrow \mu}[load_T(e)]$ is $O(\log n) w(e)$, then $E_{T \leftarrow \mu}[cost(T, \gamma)] = costA(G, \gamma) + O(\log n) costB(G, \gamma)$.*

Proof. Let f_T be the function mapping each edge $\{u, v\} \in E_T$ to a path in G connecting u and v.

Given a decomposition tree T, consider the cost of γ on T: $cost(T, \gamma) = \sum_{u,i} \phi(u, t_i)\gamma(u, t_i) + \sum_{u,v} w_T(u, v) \sum_{i,j} \gamma(u, t_i, v, t_j) d_L(t_i, t_j)$.

Let $\delta_\gamma(u, v) = \sum_{i,j} \gamma(u, t_i, v, t_j) d_L(t_i, t_j)$, which by the definition of a fractional assignment is required to be a semimetric. Consider an edge $\{u, v\} \in E_T$ which f_T maps to the path $< u = a_1, a_2, ..., a_q = v >$ in G. Then $\delta_\gamma(u, v) \leq \sum_{h=1}^{q-1} \delta_\gamma(a_h, a_{h+1})$ because $\delta_\gamma(u, v)$ is a semimetric.

So consider an edge $\{u, v\} \in E_T$ and suppose that $(U, V - U)$ is the partition of V that results from deleting $\{u, v\}$ from the decomposition tree T. Then the total weight crossing the cut in $(U, V - U)$ in G is exactly $w_T(u, v)$. We need to charge $w_T(u, v)\delta_\gamma(u, v)$ to the fractional solution δ_γ, but we could perform this charging in another way. Each edge $e \in E$, $e = \{u, v\}$, is mapped to the unique simple path in the decomposition tree connecting u and v. Let $p_T(e) =< e_1, e_2, ..., e_q >$ (where each $e_i \in E_T$) be this path, given by the sequence of its edges. Then $\sum_{\{u,v\} \in E_T} w_T(u, v)\delta_\gamma(u, v) = \sum_{e \in E} w(e) \sum_{\{a,b\} \in p_T(e)} \delta_\gamma(a, b)$, and this follows because for each edge $\{u, v\} \in E_T$, $w_T(u, v) = \sum_{e \in E : p_T(e) \ni \{u,v\}} w(e)$ for any decomposition tree. And using the above equation and the fact that $\delta_\gamma(a, b) \leq$

$\sum_{\{c,d\} \in f_T(a,b)} \delta_\gamma(c,d)$, we have

$$\sum_{\{u,v\} \in E_T} w_T(u,v)\delta_\gamma(u,v) \leq$$
$$\sum_{e \in E} w(e) \sum_{\{a,b\} \in p_T(e)} \sum_{\{c,d\} \in f_T(a,b)} \delta_\gamma(c,d) =$$
$$\sum_{\{u,v\} \in E} load_T(u,v)\delta_\gamma(u,v).$$

The equality above follows because we have that $load_T(e) = \sum_{\{u,v\} \in E_T \text{ s.t. } f_T(u,v) \ni e} w_T(u,v)$ and also $w_T(u,v) = \sum_{e \in E \text{ s.t. } p_T(e) \ni \{u,v\}} w(e)$. And so because the assignment costs are independent of the edge sets and hence are unchanged when replacing G by a decomposition tree T, we get (see Definition 3.3)

$$E_{T \leftarrow \mu}[cost(T,\gamma)] \leq costA(G,\gamma) +$$
$$\sum_{\{u,v\} \in E} E_{T \leftarrow \mu}[load_T(u,v)] \sum_{i,j} \gamma(u,t_i,v,t_j) d_L(t_i,t_j)$$
$$\leq costA(G,\gamma) + O(\log n) costB(G,\gamma).$$

So this proves that when we choose a decomposition tree T according to μ, the expected cost of the optimal solution does not increase by more than a factor of $O(\log n)$. We also need to demonstrate that the cost of the optimal solution, when we replace the graph G by a decomposition tree T, cannot decrease. We prove this iteratively. For any decomposition tree T, we will define an iterative process that, through a series of replacements, will transform the original graph G into T. We will denote $< G = G_1, G_2, ..., G_q = T >$ as the sequence of graphs in this iterative procedure. Moreover, for any fractional assignment γ, each step in this iterative procedure cannot decrease the cost of γ.

Suppose the current graph is G_i. We will define an augmentation operation, in which we choose some edge $\{u,v\}$ in G_i and we choose some sequence $< u, a_1, a_2, ..., a_r, v >$ of nodes connecting u and v. Let W be the weight of $\{u,v\}$ in G_i. This does not need to be a path in G_i. We delete the edge $\{u,v\}$ from G_i and we add edges $\{u,a_1\}, \{a_1,a_2\}, ..., \{a_r,v\}$, each with weight W. If any of these edges are already present in G_i, we instead increase the weight of the existing edge by W.

How do we choose which edges to replace by which paths? Consider any edge $\{u,v\}$ in G_i. There is a unique path connecting u and v in T; let $< u, a_1, a_2, ..., a_r, v >$ be the unique path, as a list of vertices. We delete the edge $\{u,v\}$ and augment along the sequence $< u, a_1, a_2, ..., a_r, v >$. If we perform this operation (starting with the graph G) iteratively on the current graph, we claim that the resulting weighted graph will be exactly the decomposition tree T.

To see this, consider any edge $e = \{u,v\}$ in T. Let $(U_e, V - U_e)$ be the partition of V that results from deleting e from T. Then using the definition of a decomposition tree, the weight $w_T(e)$ of e in T should be $w_G(U_e)$. We assume by induction that $w_{G_i}(U_e) = w_G(U_e)$. So suppose we perform the augmentation for some edge $fe = \{a,b\}$ that is in G_i. This edge is mapped to some path in T, and let $< a = a_1, a_2, ..., a_r = b >$ be this sequence of nodes in G. If the edge f crosses the cut $(U_e, V - U_e)$ then this sequence $< a = a_1, a_2, ..., a_r = b >$ crosses the cut exactly once. And if f does not cross the cut $(U_e, V - U_e)$ then this sequence $< a = a_1, a_2, ..., a_r = b >$ does not cross the cut $(U_e, V - U_e)$. So by induction, performing the augmentation operation on f in G_i does not change the cut value, so $w_{G_{i+1}}(U_e) = w_{G_i}(U_e)$. This implies that eventually the graph G_q we get at termination has $w_G(U_e) = w_{G_q}(U_e)$ for every e. Also, the graph G_q must be a tree, because otherwise there would be an edge in G_q that is not an edge in T so we would not be done yet, and the only edges that we add in performing an augmentation step are edges that are in T. So $G_q = T$: the edge set is the same, and the weights are also the same.

All that remains to show is that for any fractional solution γ, any step of this iterative procedure does not decrease the cost. If this is true, then by induction we obtain that $cost(T,\gamma) \geq cost(G,\gamma)$. So consider any step in the iterative procedure which deletes the edge $\{u,v\}$ and augments along the sequence $< u = a_0, a_1, a_2, ..., a_r, a_{r+1} = v >$. Let G_j and G_{j+1} be the graphs before and after applying this operation, respectively. Then using Definition 3.3 we get that $cost(G_{j+1},\gamma) - cost(G_j,\gamma) = [w(u,v)\sum_{i=0}^{r} \gamma(a_i,a_{i+1})] - w(u,v)\gamma(u,v)$. Since the definition of a fractional assignment (Definition 3.2) requires γ to be a semimetric, this difference in costs is always nonnegative. And this implies that each step in the iterative procedure that transforms G into the decomposition tree T does not decrease the cost, and so by induction $cost(T,\gamma) \geq cost(G,\gamma)$. ∎

4 Capacitated Metric Labeling for Constant k

Here we apply the results in Section 3 and we reduce the problem of solving Capacitated Metric Labeling (to within a $O(\log n)$ factor in the approximation ratio) to the problem of solving Capacitated Metric Labeling on (a polynomial number of) trees. From this, we give an $O(\log n)$-approximation algorithm for Capacitated Metric Labeling when the number of labels is constant. This extends Räcke's results on Minimum Bisection to a constant number of labels, to a more general cost function, and to arbitrary capacities for each label.

Theorem 1.1. There is a $O(\log n)$-approximation algorithm for CAPACITATED METRIC LABELING that runs in time $(n+1)^{3k} poly(n,k)$ (and does not violate label capacities).

Proof. This proof is similar to the $O(\log n)$-approximation algorithm for MINIMUM BISECTION in [26]. We will give an exact algorithm for the case in which the graph G is a tree, followed by a reduction to the tree case which costs $O(\log n)$ in the approximation.

Given any tree T, we can compute an optimal solution to a CAPACITATED METRIC LABELING instance on T in time $(n+1)^{3k} poly(n,k)$ using dynamic programming. Let w be the weight function $w : E_T \to \Re^+$ that gives the weight of each edge in T. We can root T arbitrarily at some node *root*; then for any other node u in T there is a well-defined subtree T_u rooted at u. In order to describe the dynamic program, we introduce the notion of a label configuration and the notion of when an assignment function is consistent with a label configuration.

DEFINTION 4.1. *A* label configuration g *is a function* $g : L \to \{0, 1, 2, ..., n\}$.

DEFINTION 4.2. *An assignment function on any set* Z *of vertices* $f : Z \to L$ *is* consistent with a label configuration g *if for all* $t_i \in L$, $|\{v \in Z | f(v) = t_i\}| = g(t_i)$.

Let $1_a : L \to \{0, 1\}$ map a to 1 but every other label to 0. For any node u in T, label a, and label configuration g, let $OPT(u, a, g)$ be the minimum cost of assignments f to T_u which are consistent with g and in which $f(u) = a$. In order to compute an optimal solution to a CAPACITATED METRIC LABELING instance on T, it suffices to compute $OPT(root, a, g)$ for all a, g.

Suppose that node u has r children $c_1, c_2, ..., c_r$, and furthermore that we have inductively computed $OPT(c_j, a, g)$ for all subtrees T_{c_j} rooted at children c_j of u, for all labels a, and all label configurations g. The question now is, how does one compute $OPT(u, a, g)$ for all labels a and label configurations g?

The trick is to define $T'_{u,j}$ to be the subtree consisting of u together with the *union* of the subtrees rooted at the first j children $u_1, u_2, ..., u_j$ of r. Define $OPT'_j(u, a, g)$ to be the minimum cost of an assignment f to the subtree $T'_{u,j}$ in which $f(u) = a$ and which is consistent with g. We compute $OPT'_1(u, a, g)$ for all a, g, then $OPT'_2(u, a, g)$ for all a, g, then $OPT'_3(u, a, g)$ for all a, g, ..., and finally $OPT'_r(u, a, g) = OPT(u, a, g)$ for all a, g.

$OPT'_1(u, a, g)$ is computed easily as follows. There is only one edge between u and c_1. Therefore $OPT'_1(u, a, g) = \min\{w(u, c_1) d_L(a, b) + OPT(c_1, b, g - 1_a)\}$, where the min is over all labels b, the "$g - 1_a$" appearing because one vertex (namely, u) already is assigned label a.

As there is only one edge between $T'_{u,l}$ and c_{l+1}, $OPT'_{l+1}(u, a, g)$ is computed as $\min\{w(u, c_{l+1}) d_L(a, b) + OPT'_l(u, a, g_1) + OPT(c_{l+1}, b, g_2)\}$, where the min is over label configurations g_1, g_2 satisfying $g_1 + g_2 + 1_a = g$.

What is the running time? As there are $(n+1)^k$ label configurations, and we have to iterate over g_1 and g_2 in the worst case, and there are $k(n+1)^k$ triples $OPT'_l(u, a, g)$ for a single u, the total time is $poly(n,k)(n+1)^{3k}$.

Now let G be arbitrary. Given the distribution μ on decomposition trees in Theorem 1, we can compute an optimal solution to CAPACITATED METRIC LABELING for each tree (and there are only $O(mn \log n)$ decomposition trees in the support of μ). Using the Fractional Decomposition Lemma, at least one decomposition tree will have an optimal solution that is $O(\log n)$ times the optimal solution on G. We can then take the best solution on any decomposition tree and let this solution be f. Then again using the Decomposition Lemma, $cost(G, f) \leq cost(T, f)$. And we are guaranteed that $cost(T, f) \leq O(\log n) OPT(G)$, so this implies the theorem. ∎

5 BALANCED METRIC LABELING

In this section, we apply our techniques to BALANCED METRIC LABELING, and for this problem (where l is the uniform capacity on the labels), we obtain a $O(\log l)$-approximation algorithm that violates capacities multiplicatively by $O(\log k)$. Our approach is again based on rounding the underlying graph to a decomposition tree. This is similar to the rounding approach used in [6] and [9] – but the critical difference is that we are not rounding the metric space on the labels into a tree metric, but rather rounding the graph that we wish to label into a tree. As we will see, when the underlying graph is a tree, rounding becomes easy – even easier than when the metric space on the labels is a tree metric.

We achieve a better approximation ratio, but at the price of a worse violation of capacities compared to the result in [24], which gives a $O(\log n)$-approximation ratio and but violates capacities multiplicatively by $O(\min\{\log k, l\})$.

We first give a linear programming relaxation (ULP) for BALANCED METRIC LABELING. We assume throughout this section, as in the definition of BALANCED METRIC LABELING in [24], that the semimetric d_L on the labels is uniform.

(5.1)

$$\begin{array}{lll} \min & cost(G,\gamma) & \\ \text{such that} & \gamma \text{ is a fractional assignment} & \\ & \sum_u \gamma(u,t_i) \leq l & \forall i \in [l] \\ & \sum_{v \in S} \delta_\gamma(u,v) \geq |S| - l & \forall S, u \in S. \end{array}$$

A spreading constraint, the last constraint is valid because the semimetric is uniform and each label has capacity l.

Here, we will describe the intuition behind the rounding procedure that we will use. The key point is that the function $\frac{\gamma(u,t_i,v,\cdot)}{\gamma(u,t_i)}$ is a probability distribution on labels t_j and we interpret this as the conditional probability that v is assigned label t_j, conditioned on u's being assigned label t_i. Suppose the input graph is a tree T; then there is a consistent way to realize these conditional probabilities and thus the expected cost will be exactly the cost in the linear program. We can simply root the tree arbitrarily at some root *root*, choose a label $f(root)$ for *root* according to $\gamma(root,\cdot)$, and then for each descendant v of *root* in T, we can choose the label of v according to $\frac{\gamma(root,f(root),v,\cdot)}{\gamma(root,f(root))}$. Then the probability that v is assigned any label t_i will be consistent with the marginal probability $\gamma(v,\cdot)$ and also the expected cost of the edge $(root,v)$ will be exactly $w(root,v)\sum_{t_i,t_j}\gamma(root,t_i,v,t_j)d_L(t_i,t_j)$, which is equal to the contribution of the edge $(root,v)$ to the value of the LP. So the expected cost of this rounding procedure will be exactly equal to the LP value.

All that we required was that the input graph actually be a tree T, so that we never close a cycle and run into an inconsistency problem where we can't exactly charge the expected cost of an edge to the contribution to the LP value. As long as this never happens, then the rounding procedure will exactly respect both the conditional probabilities on each edge and the marginals at each node.

But the input graph is not necessarily a tree. Yet, we can make use of the Fractional Decomposition Lemma (which is built heavily on the Hierarchical Decomposition Tree results of Räcke [26]) in order to transform an input graph G into a decomposition tree T so that the cost of the linear programming relaxation given above does not increase by more than a $O(\log n)$ factor. Then based on the fractional solution, we can choose a labeling at random so that the expected cost is exactly the value of the linear program (on the decomposition tree T), and the cost of this solution on the graph G will only be better than the cost on T (again using the Fractional Decomposition Lemma).

What we neglected to mention was that we also need to respect (or approximately respect) the label capacities. As we will see later, it is impossible to get any finite approximation ratio for the above problem and simultaneously violate the capacities by an $O((\log k)^{1/2-\epsilon})$ factor, so we cannot hope to do this in general. The above rounding procedure preserves the expected number of nodes assigned to each label, but because the rounding procedure is generated by propagating random choices (based on the fractional solution) along the edges of a tree, this rounding procedure will actually introduce quite a lot of dependence on long paths. So what we will actually need to do is to introduce limited independence (by "breaking" the rounding algorithm and randomly restarting periodically). Effectively, we are breaking up long paths of dependent random choices. By introducing limited limited independence, we will be able to obtain concentration results for the number of nodes assigned to any label.

Defintion 5.1. *We call a fractional solution γ (and the semimetric δ_γ associated with γ)* spreading *if*

$$\begin{array}{ll} \sum_u \gamma(u,t_i) \leq l & \forall i \in [l] \\ \sum_{v \in S} \delta_\gamma(u,v) \geq |S| - l & \forall S \subset V, u \in S. \end{array}$$

We will use the following standard lemma (see, for example, Naor and Schwartz [24]), which is useful in analyzing spreading semimetrics.

Lemma 5.1. *[24] If γ (and δ_γ) are spreading, then for any $\alpha > 1$ and any $v \in V$,*

$$\left|\left\{x | \delta_\gamma(v,x) \leq 1 - \frac{1}{\alpha}\right\}\right| \leq \alpha l.$$

5.1 A Cutting Procedure

We first solve the linear program (ULP) on the graph G, and then we define a cutting procedure that cuts few edges (using the spreading constraints in ULP). This gives us disconnected subproblems, each of which we reduce to a Balanced Metric Labeling instance on a tree via Räcke's hierarchical decompositions as in Section 4. Consider the following cutting procedure:

Cutting Procedure:
- Fix $\alpha = 2$, choose $\beta \in [\frac{1}{4}, \frac{1}{2}]$ uniformly at random.
- Let $< \pi_1, \pi_2, ..., \pi_n >$ be a permutation of V chosen uniformly at random.
- Set $S = V$.
- For $i = 1$ to n:
  - Set $B_i = \{x | \delta_\gamma(\pi_i, x) \leq \beta\} \cap S$.
  - Set $S = S - B_i$
- End.

The analysis of this cutting procedure follows almost exactly the analysis due to Fakcharoenphol et al. [15].

LEMMA 5.2. *$(B_1, B_2, ..., B_n)$ is a partition of V.*

Proof. Notice that $B_i \cap B_j = \emptyset$ for all $i \neq j$, because (assuming $i < j$) at step i, after choosing B_i, all nodes in B_i are removed from S. Also, any node u must occur in some B_i because if $\pi_j = u$, and if for all $i < j$, $u \notin B_i$, then u will be contained in B_j. ∎

Suppose that the semimetric δ_γ satisfies (ULP). Let $P_{u,v}$ denote the probability that the edge $\{u, v\}$ is cut, i.e., there is no i such that $u, v \in B_i$.

LEMMA 5.3. *$P_{u,v} \leq O(\log l)\delta_\gamma(u, v)$.*

Proof. We consider two cases:
Case 1: $\delta_\gamma(u, v) > \frac{1}{10}$. Here we use $P_{u,v} \leq 1$.
Case 2: $\delta_\gamma(u, v) \leq \frac{1}{10}$.

Consider the event $R^r_{u,v}$ that u is assigned to a node r (i.e., $u \in B_i$ for the i such that $\pi_i = r$) and v is not within distance β from r. Also, let $a_1, a_2, ..., a_q$ be all nodes that are within distance at most $\frac{1}{4}$ of u, and sorted so that distances to u are nondecreasing:

$$\delta_\gamma(u, a_1) \leq \delta_\gamma(u, a_2) \leq \cdots \leq \delta_\gamma(u, a_q).$$

In order for u to be assigned to node r, in the randomly chosen permutation π, a_r must occur before $a_1, a_2, ..., a_{r-1}$ because one of these other nodes a_i (for $i < r$) would claim u and u would be contained in B_i. The event that a_r occurs before $a_1, a_2, ..., a_{r-1}$ in the permutation π happens with probability $\frac{1}{r}$. Also, if u is assigned to r and v is not within distance β from r, we must have that

$$\delta_\gamma(u, a_r) \leq \beta < \delta_\gamma(v, a_r) \leq \delta_\gamma(u, v) + \delta_\gamma(u, a_r).$$

where the last equation follows from the triangle inequality. So β must be in an interval of length $\delta_\gamma(u, v)$. Because β is chosen from an interval of length $\frac{1}{4}$,

$$Pr[R^r_{u,v}] \leq \frac{1}{r}(4\delta_\gamma(u, v)).$$

Now we can bound the probability that edge $\{u, v\}$ is split by the above cutting procedure, in Case 2. If $\{u, v\}$ is split, then one of $R^r_{u,v}$ or $R^{r'}_{v,u}$ must occur because one of the nodes u, v must be assigned to a set B_i before the other node. So

$$\begin{aligned} P_{u,v} &\leq \sum_{r=1}^{q} Pr[R^r_{u,v}] + \sum_{r'=1}^{q} Pr[R^{r'}_{v,u}] \\ &\leq \sum_{r=1}^{q} \frac{8\delta_\gamma(u, v)}{r} \\ &\leq 8\delta_\gamma(u, v) \sum_{r=1}^{q} \frac{1}{r}. \end{aligned}$$

But how large can q be? We can bound q using the spreading constraints. Using Lemma 5.1, there are at most $2l$ points that are within distance $\frac{1}{2}$ of u because the semimetric δ_γ is spreading (i.e., apply Lemma 5.1 using $\alpha = 2$). So

$$P_{u,v} \leq \sum_{r=1}^{2l} \frac{8\delta_\gamma(u, v)}{r} = \delta_\gamma(u, v)O(\log l).$$

So in either Case 1 or in Case 2, $P_{u,v} \leq \max\{\delta_\gamma(u, v)O(\log l), 10\delta_\gamma(u, v)\} = O(\log l)\delta_\gamma(u, v)$. ∎

For every edge $\{u, v\}$ that is split, we will charge the cost $w(u, v)$ no matter how the labels for u and v are chosen later in the algorithm.

DEFINTION 5.2. *Let $C_{splitting}$ be the total weight of all edges $\{u, v\}$ for which the endpoints u, v are mapped to different sets B_i, B_j in the partition of V by the cutting procedure.*

DEFINTION 5.3. *Let*

$$\begin{aligned} A &= \sum_{u,i} \phi(u, t_i)\gamma(u, t_i), \\ B &= \sum_{u,v} w(u, v) \sum_{i,j} \gamma(u, t_i, v, t_j)1_{[i \neq j]} \end{aligned}$$

in the optimal solution to (ULP).

LEMMA 5.4. *$E[C_{splitting}] \leq O(\log l)B$.*

Proof.

$$\begin{aligned} E[C_{splitting}] &= \sum_{u,v} w(u, v)P_{u,v} \\ &\leq O(\log l) \sum_{u,v} w(u, v)\delta_\gamma(u, v) \\ &\leq O(\log l)B, \end{aligned}$$

by definition of $\delta_\gamma(u, v)$. ∎

Also, we will need to bound the size of any set B_i.

LEMMA 5.5. *For all i, $|B_i| \leq 2l$.*

Proof. This, again, follows from Lemma 5.1: only nodes v such that $\delta_\gamma(u, v) \leq \frac{1}{2}$ can be assigned to u by the cutting procedure, and we can apply Lemma 5.1 using $\alpha = 2$. ∎

5.2 Propagated Rounding on Trees Here we use Räcke's hierarchical decompositions to reduce to weighted trees, as in Section 4. Then we define a rounding procedure for each such tree, based on the solution to (ULP). We can delete each edge $\{u, v\}$ that has already been charged to $C_{splitting}$ in the graph G. The remaining graph is the disjoint union of the induced subgraphs $G[B_i]$. Notice that the components in the resulting graph G' are of size at most $2l$ because each set B_i is size at most $2l$. Let $G_1, G_2, ..., G_q$ be the components of G'.

For each such component $G_h = (V_h, E_h)$, we can apply Räcke's algorithm to construct a distribution μ_h on hierarchical decomposition trees of G_h. Then for each G_h, we can calculate the cost in (ULP) on G_i as $A_h + B_h$, where $A_h = \sum_{u \in V_h, i} \phi(u, t_i)\gamma(u, t_i)$ and $B_h = \sum_{\{u,v\} \in E_h} w(u, v) \sum_{i,j} \gamma(u, t_i, v, t_j) 1_{[i \neq j]}$. We can apply the second part of the Fractional Decomposition Lemma, Lemma 3.1, and so for each h, $E_{T \leftarrow \mu_h}[\sum_{u \in V_h, i} \phi(u, t_i)\gamma(u, t_i)] \leq A_h$ and

$$E_{T \leftarrow \mu_h}[\sum_{\{u,v\} \in E_T} w_T(u, v) \sum_{i,j} \gamma(u, t_i, v, t_j) 1_{i \neq j}] \leq O(\log(2l))B_h.$$

The quantities $\gamma(u, t_i)$ and $\gamma(u, t_i, v, t_j)$ are known, because these are just the values of the (optimal) fractional solution that results from solving (ULP) on the graph G. So since we know these values, for each G_h, we can choose a decomposition tree T_h for which the cost (against the fractional variables $\gamma(u, t_i)$ and $\gamma(u, t_i, v, t_j)$) is at most the expectation when T_h is sampled from μ. If we choose one such decomposition tree for each G_h, the cost of this disjoint union of trees T_h (against the fractional variables $\gamma(u, t_i)$ and $\gamma(u, t_i, v, t_j)$) is at most $A + O(\log l)B$.

We can now define a rounding procedure for any tree T_h. Fix any root $r \in T_h$.

Rounding Procedure for T_h:
- Choose a label for r according to $\gamma(r, \cdot)$.
- While there is an unlabeled node $u \in T_h$ that is adjacent to a labeled node $v \in T_h$:
  - Choose a label $f(u)$ for u according to the distribution $\frac{\gamma(u, \cdot, v, f(v))}{\gamma(v, f(v))}$.

We can interpret $\frac{\gamma(u, t_i, v, f(v))}{\gamma(v, f(v))}$ as the conditional probability of choosing $f(u) = t_i$ conditioned on v's being labeled $f(v)$. Note that $\sum_i \gamma(u, t_i, v, t_j) = \gamma(v, t_j)$, so this is actually a probability distribution.

Lemma 5.6. *[Rounding Lemma] Let f be a random assignment produced by the above Rounding Procedure for T_h. Then $E_f[cost(T_h, f)] = cost(T_h, \gamma)$.*

Note that f need not be a valid assignment, i.e., some random choices of f can result in too many nodes' being mapped to a particular label, and thereby exceeding the capacity of that label.

Proof. Consider a node $u \in T_h$, and consider the unique root-to-leaf path $< r, u_1, u_2, ..., u_p, u >$. The above Rounding Procedure will first label r, and then u_1, followed by u_2, ..., until u is labeled. We need to prove that $Pr_f[f(u) = t_i] = \gamma(u, t_i)$; this will imply that the Rounding Procedure respects the first-order probabilities in the linear programming solution γ. We prove this by induction, on the number of nodes in the path $< r, u_1, u_2, ..., u_p, u >$. By construction $Pr_f[f(r) = t_i] = \gamma(r, t_i)$, so we next assume that $Pr_f[f(u_s) = t_i] = \gamma(u_s, t_i)$ for all i, for a particular s. Then consider u_{s+1}. We can write out the probability that u_{s+1} is assigned label j using the Law of Total Probability:

$$Pr_f[f(u_{s+1}) = t_j] = \sum_i Pr_f[f(u_s) = t_i] Pr_f[f(u_{s+1}) = t_j | f(u_s) = t_i].$$

Using the inductive hypothesis that $Pr_f[f(u_s) = t_i] = \gamma(u_s, t_i)$, we get

$$Pr_f[f(u_{s+1}) = t_j] = \sum_i \gamma(u_s, t_i) Pr_f[f(u_{s+1}) = t_j | f(u_s) = t_i].$$

And then substituting in the probability distribution used by the Rounding Procedure for u_{s+1}, we get

$$\begin{aligned} Pr_f[f(u_{s+1}) = t_j] &= \sum_i \gamma(u_s, t_i) \frac{\gamma(u_{s+1}, t_j, u_s, t_i)}{\gamma(u_s, t_i)} \\ &= \sum_i \gamma(u_{s+1}, t_j, u_s, t_i) \\ &= \gamma(u_{s+1}, t_j). \end{aligned}$$

We also need to prove that the Rounding Procedure also respects second order probabilities in the linear programming solution, i.e., for adjacent $u, v \in T_h$, $Pr[f(u) = t_i, f(v) = t_j] = \gamma(u, t_i, v, t_j)$, because the expected cost of the Rounding Procedure on the edge $\{u, v\}$ is a function of these second order probabilities (it is exactly $\sum_{t_i, t_j} \gamma(u, t_i, v, t_j) 1_{i \neq j}$). Assume without loss of generality that u occurs on the unique root-to-leaf path to v, so that u is labeled immediately before labeling v. Then using the above inductive argument, $Pr_f[f(u) = t_i] = \gamma(u, t_i)$. Then the probability that we now choose label t_j for v is exactly $\frac{\gamma(u, t_i, v, t_j)}{\gamma(u, t_i)}$ and this implies the statement.

So the rounding algorithm respects both the first order and second order probabilities in the linear program, and so the expected cost of each node and each edge when using the above Rounding Procedure is exactly what the corresponding cost is in the linear program. ∎

We are now ready to give a bicriteria approximation algorithm for BALANCED METRIC LABELING. We need a tail bound from Schmidt, Siegel and Srinivasan [27]:

LEMMA 5.7. *[27] Let S be the sum of p independent random variables, each of which is confined to $[0, 1]$. Let $\mu = E[S]$. Then if $\delta \geq 1$ and $k \geq \delta\mu$, then*

$$Pr[|S - \mu| \geq \delta\mu] \leq \Big(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\Big)^{\mu}.$$

Algorithm 1
- Solve (ULP).
- Run the **Cutting Procedure** on this solution.
- For each component C_h that results:
  - Apply Theorem 2.1 to sample a decomposition tree T_h.
- End
- For each decomposition tree T_h:
  - Root each T_h arbitrarily.
  - Run **Rounding Procedure for T_h.**
- End.
- Output f.

Theorem 1.2. Suppose f is generated at random by running **Algorithm** 1. Then

$$E_f[cost(G, f)] \leq O(\log l)OPT(G).$$

With high probability, simultaneously all labels t_i satisfy $|\{u|f(u) = t_i\}| \leq O(\log k)l$.

Proof. We can bound the cost of **Algorithm** 1 by

$$C_{splitting} + \sum_h E_f[cost(T_h, f)].$$

From Lemma 5.4 we get that $E[C_{splitting}] \leq O(\log l)B$. Using the Rounding Lemma, Lemma 5.6,

$$E_f[cost(T_h, f)] = cost(T_h, \gamma).$$

And then $\sum_h E_f[cost(T_h, f)]$ is just $\sum_h cost(T_h, \gamma)$. Using the Fractional Decomposition Lemma and the fact that each component C_h is of size at most l, we get that $\sum_h cost(T_h, \gamma) = A + O(\log l)B$ because we can apply the Fractional Decomposition Lemma separately for each component C_h. This implies the bound on the expected cost of **Algorithm 1**. We also need to bound the probability that label t_i receives too many nodes. So consider the random variable $X_{h,i}$ which is defined to be the number of nodes in C_h that are mapped to t_i in f. Then using the fact that the **Rounding Procedure for** T_h respects first order probabilities in the linear programming solution, i.e., $Pr_f[f(u) = t_i] = \gamma(u, t_i)$, and this implies that $E[X_{h,i}] = \sum_{u \in C_h} \gamma(u, t_i)$. In particular, set $Y_{h,i} = \frac{X_{h,i}}{2l}$. Then because the size of any component is bounded by $2l$, each variable $Y_{h,i}$ is bounded in $[0, 1]$ and set $S = \sum_h Y_{h,i}$. Then $E[S] = \frac{1}{2l}\sum_h E[X_{h,i}] \leq \frac{1}{2l}l$ using the constraint that $\sum_u \gamma(u, t_i) \leq l$ for all $t_i \in L$. So using Lemma 5.7,

$$Pr[S > O(\log k)] \leq \frac{1}{k^c}.$$

So this implies that with high probability $\sum_h X_{h,i} \leq (2l)O(\log k)$ simultaneously for all labels t_i (applying the above inequality and a union bound over all k labels). ∎

5.3 Integrality Ratio for CAPACITATED METRIC LABELING ON THE LINE In this section we show an $\Omega(n^{1/5})$-integrality ratio for the natural generalization of the linear programming relaxation (ULP) for BALANCED METRIC LABELING to the case in which the label semimetric d_L is a line. We will refer to this problem as CAPACITATED METRIC LABELING ON THE LINE. The LP that we consider here is the same as in the case for BALANCED METRIC LABELING, except for the spreading constraint. We repeat the LP for BALANCED METRIC LABELING:

$$\begin{array}{rll} \min & cost(G, \gamma) & \\ \text{such that} & & \\ & \gamma \text{ is a fractional assignment} & \\ & \sum_u \gamma(u, t_i) \leq l & \forall i \in [l] \\ & \sum_{v \in S} \delta_\gamma(u, v) \geq |S| - l & \forall S, u \in S. \end{array}$$

Note that given a fractional solution γ (Definition 3.2), the term $cost(G, \gamma)$ (Definition 3.3) and the semimetric δ on the nodes in G depend on the label semimetric implicitly. So we do not need to rewrite these definitions when we instead consider CAPACITATED METRIC LABELING ON THE LINE.

If the label space is a line, then we can write a stronger spreading constraint than $\sum_{v \in S} \delta_\gamma(u, v) \geq |S| - l$ which will still be valid for any feasible assignment. Instead, we use the spreading constraint:

$$\forall S, u \in S \sum_{v \in S} \delta(u, v) \geq l + 2l + 3l + \cdots + l\left\lfloor \frac{|S| - l}{l} \right\rfloor.$$

The graph in our example is a line with n nodes. The label space is a line metric with $k = n^5$ labels. For each node u, the assignment cost $\phi(u, t_i)$ is zero for exactly n labels. For all other labels the assignment cost is infinite. In particular, for each u flip a set of n coins such that for each coin the probability that it is heads is $1/\sqrt{n}$. If the rth coin is heads then $\phi(u, l_{(r-1)n^4+1}) = 0$ else $\phi(u, l_{(r-1)n^4+2u}) = 0$. In other words, this defines the rth label that node u can be assigned to. All labels have capacity 1.

Let $B_i = \{u : \phi(u, t_i) = 0\}$, i.e., B_i is the set of nodes that could be mapped to label t_i.

LEMMA 5.8. *With high probability,* $2\sqrt{n} \geq |B_{(r-1)n^4+1}| \geq \sqrt{n}/2$ *for all* r.

Proof. The lemma follows immediately from the fact that $|B_{(r-1)n^4+1}|$ is distributed according to the binomial distribution with n trials and success probability $1/\sqrt{n}$. ∎

Set $D_{u,v}$ equal to $\{r : \phi(u, l_{(r-1)n^4+2u}) = 0$ and $\phi(v, l_{(r-1)n^4+2v}) = 0\}$.

LEMMA 5.9. *With high probability* $|D_{u,v}| \geq \frac{n}{2}$ *for all* u, v.

Proof. Follows immediately from the fact that $|D_{u,v}|$ is distributed according to the binomial distribution with n trials and success probability $(1 - \frac{1}{\sqrt{n}})^2$. ∎

We assume that from now on that the conditions of the previous two lemmas are satisfied.

LEMMA 5.10. *For any integral solution, either the objective is more than* $n^4/2$ *or some label is assigned to at least* $\sqrt{n}/2$ *nodes.*

Proof.

- Case 1: There are two nodes u, v such that u is assigned its rth label and v is assigned its r'th label for $r \neq r'$. The distance between these two assigned labels is at least $n^4/2$.
- Case 2: There exists some r such that all nodes are assigned their rth label. In this case all the nodes in $B_{(r-1)n^4+1}$ are assigned label $l_{(r-1)n^4+1}$. This means that the capacity of label $l_{(r-1)n^4+1}$ is violated by at least a factor of $\sqrt{n}/2$.

∎

Now consider the fractional solution in which $\gamma(u, t_i, v, t_j) = 1/n$ if and only if there is an r such that t_i is the rth label for node u and t_j is the rth label for node v. Set $\gamma(u, t_i, v, t_j) = 0$ otherwise. We will call this the *Canonical Fractional Solution.*

LEMMA 5.11. *The* Canonical Fractional Solution *is a feasible fractional solution of value at most* n^3 *for which no capacities are violated.*

Proof. If $\gamma(u, t_i, v, t_j) = 1/n$ then $|i - j| \leq n$. Therefore $\sum_{u,v} \sum_{i,j} \gamma(u, t_i, v, t_j) \leq n^2 \cdot n = n^3$. The total (fractional) number of nodes that are assigned any label is bounded by $\max_i |B_i|/n \leq 2\sqrt{n}/n \leq 1$. Hence the capacities are respected. Last, for any node u and any node set S, we have

$$\sum_{v \in S} \delta(u, v) \geq \sum_{v \in S} |2u - 2v| D_{u,v}/n \geq \sum_{v \in S} |u - v| \geq 1 + 2 + \cdots + |S - 1|.$$

∎

If we denote the size of the instance by n then n is $O(n^5)$. The above shows that the integrality ratio is $\Omega(n)$. Hence we have

THEOREM 5.1. *The integrality gap of the natural spreading semimetric LP relaxation of the* CAPACITATED METRIC LABELING ON THE LINE *is* $\Omega(n^{1/5})$.

6 Congestion and Bicriteria Hardness

Here we study the inapproximability of CAPACITATED METRIC LABELING. We first prove that it is impossible to approximate the value of an instance of CAPACITATED METRIC LABELING to within any bounded factor if one is not allowed to violate capacities, if $P \neq NP$. We accomplish this through a reduction from 3-PARTITION. Yet, this does not address the more interesting question of how hard CAPACITATED METRIC LABELING is to approximate when we are allowed to violate capacities.

To study this question, we introduce the notion of the congestion: Our hardness results are inspired by those obtained in [3], [2] and [10] and we introduce a similar notion of congestion relevant in the context of CAPACITATED METRIC LABELING. Through studying congestion, we are able to prove that any polynomial-time approximation algorithm that achieves a finite approximation ratio must multiplicatively violate the label capacities by $\Omega((\log k)^{1/2-\epsilon})$. Using our terminology, CAPACITATED METRIC LABELING is $(\infty, \Omega((\log k)^{1/2-\epsilon}))$-bicriteria hard (assuming $NP \not\subseteq ZPTIME(n^{polylog\ n})$).

We also study the question of giving an approximation algorithm for the congestion of a CAPACITATED METRIC LABELING INSTANCE. The main open question in this work is whether or not there is an approximation algorithm that simultaneously achieves a polylogarithmic approximation in cost and a polylogarithmic violation of label capacities in general. As we explain in this section, any such approximation algorithm would also yield a polylogarithmic approximation algorithm for congestion and we study this question direction and give such an approximation algorithm: we give a rounding procedure that gives a $O(\log k)$ approximation algorithm for determining the congestion of an instance of Capacitated Metric Labeling. Our technique can be regarded as an analogue of the randomized rounding techniques for flows.

6.1 Reduction from 3-Partition **to** Capacitated Metric Labeling In this section, we give a reduction from 3-Partition to Capacitated Metric Labeling. Recall that in 3-Partition, given are a (multi-)set $S = \{a_1, a_2, \ldots, a_n\}$ of $n = 3m$ positive integers, and the goal is to partition S into m subsets (bins) $S_1, S_2, \ldots, S_m$ such that the sums of the numbers in the subsets are equal. Let B denote the (desired) sum of each subset S_i, or equivalently, let the total sum of the numbers in S be mB. 3-Partition remains NP-complete when every integer in S is strictly between $B/4$ and $B/2$. In this case, each subset S_i is forced to consist of exactly three elements. We note that 3-Partition is strongly NP-complete, i.e., the problem remains NP-complete even when the integers in S are bounded above by a polynomial in n. Say this polynomial is n^α for the constant $\alpha > 0$. See Garey and Johnson [17] for more details on 3-Partition.

Theorem 6.1. *It is NP-hard to approximate* Capacitated Metric Labeling *to within a factor $n^{1-\epsilon}$, for arbitrary fixed $\epsilon > 0$ when all capacities are ones (and cannot be violated) and semimetric d_L is a unweighted line metric (i.e., unevenly spaced points on the line).*

Proof. As mentioned before, we use a reduction from 3-Partition. First we consider the weighted line metric case. The graph G in the Metric Labeling instance is just a set $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$ of disjoint paths such that path P_i, $1 \leq i \leq n$, has length $a_i - 1$ (and thus has a_i vertices). Metric d_L is a line metric obtained by concatenation of m identical paths $Q =< q_1, q_2, \ldots, q_B, q_{B+1} >$ in which the length of edge $\{q_B, q_{B+1}\}$ is ∞ and all other lengths are ones; the triangle inequality is satisfied. All assignment costs are zeroes.

When the instance of 3-Partition has a solution, we consider each copy of path Q corresponding to a bin S_i, $1 \leq i \leq m$, and we map paths $P_j, P_{j'}, P_{j''}$ corresponding to integers $a_j, a_{j'}, a_{j''}$ that we pack in bin S_i into labels $q_1, q_2, \ldots, q_B$ in a straightforward manner with total cost $B - 3$. Thus in this case the total cost of our Capacitated Metric Labeling instance is $m(B-3) \leq n^{\alpha+1}$. On the other hand, when the instance of 3-Partition does not have a solution, due to our construction, we need to span at least one edge $\{q_B, q_{B+1}\}$ in our line metric over labels and thus we incur a cost of ∞. Thus there is no finite approximation factor for Capacitated Metric Labeling unless P=NP.

Now consider the case in which metric d_L is an unweighted line metric. We do exactly as aforementioned except that path Q is $< q_1, q_2, \ldots, q_B, q_{B+1}, \ldots, q_{B+1+n^h} >$, for a large enough h to be determined later. Now the assignment cost is ∞ to labels $q_{B+1}, \ldots, q_{B+n^h}$ for all aforementioned vertices in G and zero otherwise. We also add $k - mB$ new dummy vertices (with no edges) to G whose assignment costs are zero to labels $q_{B+1}, \ldots, q_{B+n^h}$ and ∞ otherwise. Note that in this case, our number k of labels, $k = m(B + n^h + 1)$, is also the number of vertices in G. The case in which the instance of 3-Partition has a solution is the same as before. However when the instance has no solution, due to our construction, we now need to span one edge $e \in E(G)$ over a subpath $q_{B+1}, \ldots, q_{B+1+n^h}$ in our unweighted line metric and thus we incur a cost of at least n^h. Thus in this case if $n^h > n^{(1-\epsilon)(\alpha+h+2)} n^{\alpha+1} \geq (\frac{n}{3}(n^\alpha + n^h + 1))^{(1-\epsilon)} n^{\alpha+1} \geq |V(G)|^{1-\epsilon} n^{\alpha+1}$, i.e., $h > \frac{\alpha(2-\epsilon)+3-2\epsilon}{\epsilon}$, there is no $|V(G)|^{1-\epsilon}$ approximation for Capacitated Metric Labeling unless P=NP. ∎

It is worth mentioning that in the case of the weighted line metric above, there is no need of assignment restrictions, i.e., all assignment costs are zeroes. This is indeed the linear arrangement problem when the underlying line (corresponding to order) is weighted.

6.2 Bicriteria Hardness The congestion of an instance of Capacitated Metric Labeling is the minimum value C so that if we were to scale up the label capacities by a factor of C, there would be a valid assignment that has zero cost.

Defintion 6.1. *Given G, ϕ, and d_L, and label capacities $< l_1, l_2, ..., l_k >$, we define the* congestion *of the corresponding* Capacitated Metric Labeling *instance to be the smallest value of C such that scaling up capacities by C has a solution of zero cost (and if no such value of C exists, then we will say that the congestion is ∞).*

Congestion is a framework in which to study the question, how much must a polynomial-time approximation algorithm cheat in order to obtain any finite approximation ratio for Capacitated Metric Labeling? We prove that (under certain complexity assumptions) there is no polynomial-time approximation algorithm that can approximate the congestion to within $O((\log k)^{1/2-\epsilon})$ on every instance. And this yields an answer to the above question, as follows: any polynomial-time approximation algorithm that achieves a finite approximation ratio must multiplicatively violate the label capacities by $\Omega((\log k)^{1/2-\epsilon})$, as we will demonstrate in this section.

Theorem 1.3. If $NP \not\subseteq ZPTIME(n^{polylog\ n})$, then for all $\epsilon > 0$, there is a function $h(k)$ which is $\Omega((\log k)^{1/2-\epsilon})$ such that no polynomial-time approximation algorithm can approximate the congestion of a Capacitated Metric Labeling instance by a factor of $h(k)$, when $|L| = k$. This holds even if the distances in the label semimetric are restricted to be zero or one.

Proof. Fix $\epsilon > 0$. Our reduction is from [4], which considered questions about coloring paths in graphs because of applications to wavelength assignment problems in routing. Formally, the results of [4] which we need are these. Suppose we have a set of demands which are given as paths through an unweighted, n-node network $G = (V, E)$ with m edges. Suppose we want to color these demands with μ colors $1, 2, ..., \mu$; let the color of the associated path equal that of the demand associated with it. Then, assuming $NP \not\subseteq ZPTIME(n^{polylog\ n})$, it is impossible in polynomial time to distinguish between the case in which all demands can be colored such that there is at most one demand path of each color passing through each edge, and the case in which for any coloring there is some edge with $\Omega((\log m)^{1/2-\epsilon})$ paths of the *same* color passing through it. Furthermore, the same result holds with "$\Omega((\log m)^{1/2-\epsilon})$" replaced by "$\Omega((\log \mu)^{1/2-\epsilon})$."

We reduce to Capacitated Metric Labeling as follows. The graph in the instance, now called $G' = (V', E')$, will be a vertex-disjoint collection of unweighted paths. For each demand path in G, with, say, h edges, build a path in G' with h *nodes*. The label set will be $E \times [\mu]$, i.e., $\{(e, c) | e \in E, 1 \le c \le \mu\}$, of size $k = m\mu$. Every label capacity is 1. The distance between (e_1, c_1) and (e_2, c_2) is 0 if $c_1 = c_2$ and 1 otherwise.

The only thing left to do is to choose assignment costs ϕ. Choose a direction for each of the demand paths in G. Each node v' in V' corresponds in a natural way to an edge $edge(v')$ in G: specifically, the lth *node* in the path in G' corresponding to demand path d in G corresponds to the lth *edge* in the dth demand path in G. (We needed to orient the paths so that we could talk about the "lth edge" in the path.) Now for any node $v' \in V'$, define $\phi(v', (e, c))$ to be 0 if $e = edge(v')$ and 1 otherwise. In other words, a node v' in V' can be assigned only to a label whose first component is the edge in E from which v' arose. This completes the reduction.

Now we prove that if all demands can be colored such that in G there is at most one path of each color passing through each edge, then the congestion of the Capacitated Metric Labeling instance is 1, and if for any coloring there is some edge in G with $\Omega((\log m)^{1/2-\epsilon})$ paths of the same one color passing through it, then the congestion is $\Omega((\log m)^{1/2-\epsilon})$.

Suppose we have a color $color(d)$ for each demand path d so that, in G, at most one demand path of each color passes through each edge. For any node v' in V' corresponding to, say, demand path d, assign v' to label $(edge(v'), color(d))$. Note that the total assignment cost is 0, and that, since all adjacent nodes in G' get labels with the same second component, the total label cost is 0. Since there is at most one demand path of each color passing through each edge, the unit capacities are satisfied. Hence the congestion is 1.

Now suppose that for any coloring of demand paths there is some edge in G with at least $\alpha \in \mathbb{N}$ paths all of the same color passing through it. Any assignment f of zero total cost must incur 0 assignment cost, so that $f(v') = (e, c)$ implies that $e = edge(v')$, and 0 label cost, so that if v', v'' are on the same path in G', then the second components of the labels they get must be the same, i.e., all the nodes on one path in G' get the same color. If it were possible to inflate the capacities, which are now all 1, by a factor of $\alpha - 1$ and still get zero cost, then it would be possible to assign a color to each path in G' such that for every edge e of G, for any color c, the number of $v' \in V'$ with $f(v') = (e, c)$ would be at most $\alpha - 1$, and hence the number of demand paths of color c passing through e would be at most $\alpha - 1$, thereby contradicting our assumption. Hence the congestion is at least α, which is $\Omega((\log m)^{1/2-\epsilon})$.

If $m \ge \mu$, then we are done, since in that case $m \ge \sqrt{k}$ and hence we have $\Omega((\log k)^{1/2-\epsilon})$-hardness. Otherwise, we use the fact that [4] also proves $(\log \mu)^{1/2-\epsilon}$ hardness, from which we again get $\Omega((\log k)^{1/2-\epsilon})$ hardness. This completes the proof. ∎

The above theorem implies:

Corollary 1.1. If $NP \not\subseteq ZPTIME(n^{polylog\ n})$, then there is no polynomial-time approximation algorithm for Capacitated Metric Labeling that achieves a finite approximation ratio in the cost and violates the capacities of labels multiplicatively by $O((\log k)^{1/2-\epsilon})$. This holds even if the distances in the label semimetric are restricted to be zero or one.

Proof. Suppose we are given a polynomial-time approximation algorithm that achieves some finite approximation ratio compared to the optimal solution, and violates the capacity of each label by at most a multiplicative factor of C. From this approximation algorithm, we can obtain a polynomial-time algorithm for distinguishing between the case in which there is a valid assignment of zero cost, and the case in which there is no assignment of zero cost even if the label capacities are scaled up by a factor of C.

We can simply run the given approximation algorithm on an instance of Capacitated Metric Labeling, and if it returns an assignment of zero cost, then we output "YES," and if not, we output "NO." Using the observation in Section 1.2 that there are only a polynomial number of possible values for the congestion, we obtain a C-approximation algorithm for determining the congestion of a Capacitated Metric Labeling instance, and this contradicts Theorem 1.3. ∎

6.3 Approximating Zero-Cost Solutions Here we consider the problem of approximating the congestion of an instance of Capacitated Metric Labeling. We give a $O(\log k)$ approximation algorithm for this problem. In designing an approximation algorithm, we first choose a natural rounding algorithm for rounding a fractional assignment into an integral assignment. The fractional assignment that we eventually round will be the output from solving an LP. But actually, since we have already fixed our rounding procedure, we introduce a new constraint in our LP that ensures that we only consider fractional solutions on which the rounding procedure will perform well. This constraint will be valid for the actual assignments, and is based on making sure that for the fractional solution, for each possible choice of our already-chosen rounding procedure, the expected number of nodes assigned to any label cannot change by too much. To the best of our knowledge, this technique is novel and may be useful in other contexts as a way of planning for a particular rounding procedure to ensure tight concentration.

We introduce the notion of a partition-uniform semimetric, which will be useful in describing our approximation algorithm for congestion.

Defintion 6.2. *A function $d_L : L \times L \to \Re$ is a* partition-uniform semimetric *if d_L is a semimetric and there is a partition $(T_1, T_2, \dots, T_q)$ of L such that*

$$d_L(a,b) = \begin{cases} 0 & \exists i \text{ such that } a,b \in T_i \\ 1 & \text{else.} \end{cases}$$

Our approach to designing an approximation algorithm for determining the congestion of a Capacitated Metric Labeling instance is to round an appropriate linear program. (Surprise!) We first give some simple observations on solutions that have zero cost. If we are interested in a solution of zero cost, then because ϕ is always nonnegative, we can restrict $\gamma(u, t_i)$ to be zero whenever $\phi(u, t_i) > 0$. Because we can assume all edges in G have nonzero weight (we can just delete edges of zero weight), no nodes u, v that are adjacent in G are ever assigned to T_i, T_j respectively for $i \neq j$. (If nodes u and v were assigned to T_i, T_j, then the edge $\{u, v\}$ would be charged some positive cost.)

Hence we can decompose G into connected components $C_1, C_2, ..., C_r$. We can assign all nodes in a component at once to a particular set in the part T_i. Then once we have fixed this set, we can later decide which nodes in C_j get mapped to which labels in T_i. This strategy will be the basis for our linear program.

Defintion 6.3. *Let $F \subseteq V \times L$ be the set of all (node, label) pairs for which $\phi(u, t_i) = 0$. Let $F_u \subseteq L$ be $F \cap (\{u\} \times L)$, i.e., the set of all labels t_i to which node u has zero assignment cost.*

Defintion 6.4. *We call γ a* zero-cost fractional solution *if*

$$\begin{aligned} \textstyle\sum_{j: t_j \in F_u} \gamma(u, t_j) &= 1 && \forall u \\ \gamma(u, t_j) &= 0 && \forall u \forall t_j \notin F_u \\ \textstyle\sum_u \gamma(u, t_j) &\leq l_j && \forall t_j \in L \\ \gamma(u, t_j) &\geq 0 && \forall u \in V, t_j \in L. \end{aligned}$$

Additionally, we will need a notion of consistency for zero-cost solutions, i.e., if $\gamma(u, t_i) \neq 0$ for some $t_i \in T_j$, then for any other v contained in the same connected component, v must have nonzero probability of being mapped to some other label in T_j. Otherwise there is no way to find a distribution on assignments which is both consistent with $\gamma(u, \cdot)$ and $\gamma(v, \cdot)$ and has zero cost in expectation (and hence zero cost for each assignment that occurs with positive probability).

Defintion 6.5. *We call a zero-cost solution γ* consistent *if there is an extension of γ to $C \times T$ (where $C = \cup_h \{C_h\}, T = \cup_i \{T_i\}$) such that*

$$\begin{aligned} \textstyle\sum_{T_i} \gamma(C_h, T_i) &= 1 && \forall h \\ \textstyle\sum_{t_j \in T_i} \gamma(u, t_j) &= \gamma(C_h, T_i) && \forall h, i, \forall u \in C_h \\ \gamma(C_h, T_i) &\geq 0 && \forall h, i. \end{aligned}$$

The second condition enforces that each component can be regarded as being assigned as a group.

A first attempt at a rounding algorithm is to perform an independent choice for each component C_i. Unfortunately, with just the restrictions given above, this can fail. It could be the case that once one chooses which set T_a to map to, every node in C_i must be mapped to the same label in T_a, i.e., for all $u \in C_i, F_u \cap T_a = \{t_j\}$. Such a situation could be possible, while still satisfying the capacity constraints in expectation, because the probability that C_i is assigned to any T_a could be quite small, but for *every* such T_a that it can be assigned to, this situation could arise. Thus the expectation is bounded, but conditioned on any choice of which T_a to map to, the conditional expectation of *some* label T_i could be as large as $|C_i|$.

The key insight is that one can also enforce that the conditional expectation of the number of nodes assigned to a particular label t_i cannot change by too much, for each choice. Of course, this constraint is valid because in an integral assignment, each component C_i is mapped to exactly one set T_a of labels. So after fixing that each component is mapped to the appropriate set T_a of labels, one has not changed the conditional expectation of the number of nodes mapped to any label precisely because this was the only choice that one could make! This constraint on the conditional expectation is actually a linear constraint on the variables, and will allow us to enforce a condition that implies concentration, and this is how one obtains our result.

Consider the following linear program:

$$\begin{array}{rl} \min & cost(G,\gamma) \\ \text{such that} & \gamma \text{ is a zero-cost solution;} \\ & \gamma \text{ is consistent;} \\ \forall C_h, T_j, \forall t_i \in T_j & \sum_{u \in C_h} \gamma(u,t_i) \le l_i \gamma(C_h, T_j). \end{array}$$

As we noted, the variable $\gamma(C_h, T_i)$ is interpreted as the probability that component C_h is mapped to set T_i. We will perform a two-round rounding procedure on a fractional solution, by first choosing for each component which set in the partition to which this set is mapped. Every node in the same component must be mapped to the same set T_i in order for the solution to have zero cost. In the second round, we decide which nodes in C_h are assigned to which labels in T_i.

We will refer to the last constraint in the LP as the Conditional Expectation Constraint. It can be easily verified that any integral solution that has zero cost satisfies this constraint. This constraint will allow us to get large deviation bounds on the number of nodes assigned to a label t_i. We can divide through this constraint by $\gamma(C_h, T_i)$. If we assign C_h to T_i, then we need to update the conditional probabilities $\gamma(u, \cdot)$. In particular, for any label $t_j \notin T_i$ and for any node $u \in C_h$, the probability that u is assigned to t_j (conditioned on C_h's being assigned to T_i) is zero, because node u and all nodes in C_h have already decided on T_i.

The probability that node u is assigned to a label $t_j \in T_i$ is no longer $\gamma(u, t_j)$, because now there are many labels t_h for which $\gamma(u, t_h)$ is positive but to which u can no longer be assigned. So in order to construct a conditional probability distribution, we need to divide by $\gamma(C_h, T_i)$, i.e., conditioned on C_h's being assigned to T_i, the probability that u is assigned to label $t_j \in T_i$ is now $\frac{\gamma(u,t_j)}{\gamma(C_h,T_i)}$. And using the consistency condition on γ, this will be a probability distribution.

In order to get a large deviation bound for the number of nodes mapped to a particular label, we need that the change in the expected number of nodes mapped to a particular label (after the first round of the rounding procedure) is not too large. Initially, the expectation is at most l_j for any label t_j. The Conditional Expectation Constraint gives us a bound on how much the conditional expectation of the number of nodes mapped to t_j can change after specifying that C_h is assigned to set T_i.

We can now describe the full rounding procedure:

Two-Level Rounding Procedure:

- For each component C_h :
  - Set $f_1(C_h) = T_i$ where T_i is sampled from the distribution $\gamma(C_h, \cdot)$.
- For each component C_h:
  - For each node $u \in C_h$:
    - Set $f_2(u) = t_j$ where t_j is sampled from the distribution $\frac{\gamma(u,\cdot)}{\gamma(C_h,T_i)}$
- Output f_2.

LEMMA 6.1. *The **Two-Level Rounding Procedure** incurs zero cost.*

Proof. By construction, a node u can only be assigned to a label t_j for which $\gamma(u, t_j)$ is positive. So from the constraint that γ be of zero cost, no node is assigned to any label for which it has positive assignment cost. Also, consider any adjacent nodes u, v. These nodes are in the same component C_h, and by design all nodes in this component are mapped to the same set T_i. So now no matter where these two nodes are mapped within T_i (for any i), the distance from $f_2(u)$ to $f_2(v)$ is zero, because d_L is a partition-uniform semimetric and T_i is a part in the partition. ∎

LEMMA 6.2. *If there is a* CAPACITATED METRIC LABELING *solution in which each label t_j is assigned at most l_j labels, then the above linear program is feasible.*

Proof. Set $\gamma(u, t_j) = 1$ iff u is assigned to label t_j in the optimal solution. Also set $\gamma(C_h, T_i) = 1$ iff C_h is assigned to T_i. (As noted earlier, C_h must be assigned to a particular set T_i in any zero-cost solution, so this is well defined.) Then for these choices of γ, the only nontrivial condition to check is the Conditional Expectation Constraint. If $\gamma(C_h, T_i) = 0$ then for all $t_j \in T_i$ and all $u \in C_h$, $\gamma(u, t_j) = 0$. If $\gamma(C_h, T_i) = 1$ then $\sum_{u \in C_h} \gamma(u, t_j)$ is just the total number of nodes assigned to label t_j, which is at most l_j because the optimal assignment is a valid assignment. ∎

Lemma 6.3. (Conditional Expectation Lemma)

$$E\Big[|\{u|f_2(u) = t_j\}|\Big|f_1(C_h) = T_i\Big] \leq E\Big[|\{u|f_2(u) = t_j\}\Big] + l_j.$$

Proof. If we fix $f_1(C_h) = T_i$, a node $u \in C_h$ can only be mapped to labels in T_i. So we need to renormalize the probability of u being mapped to each label in T_i so that this is a probability distribution, and $\sum_{t_j \in T_i} \gamma(u, t_j) = \gamma(C_h, T_i)$ using the definition of a consistent zero-cost solution in Definition 6.5. So this implies that after we fix $f_1(C_h) = T_i$, the probability of $u \in C_h$ being mapped to $t_j \in T_i$ is $\frac{\gamma(u,t_j)}{\gamma(C_h,T_i)}$. So $Pr[f_2(u) = t_j|f_1(C_h) = T_i] = \frac{\gamma(u,t_j)}{\gamma(C_h,T_i)}$. Then from this expression we can deduce that

$$E[|\{u|f_2(u) = t_j\}|f_1(C_h) = T_i] - E[|\{u|f_2(u) = t_j\}] = \sum_{u \in C_h} \Big(\frac{\gamma(u,t_j)}{\gamma(C_h,T_i)} - \gamma(u,t_j)\Big) \leq \sum_{u \in C_h} \Big(\frac{\gamma(u,t_j)}{\gamma(C_h,T_i)}\Big).$$

And using the constraint in the linear program, we have that $\sum_{u \in C_h} \frac{\gamma(u,t_j)}{\gamma(C_h,T_i)} \leq l_j$, and this yields the lemma. ∎

Lemma 6.4. *After the first-level rounding, with high probability for any label t_j, the conditional expectation of the number of nodes mapped to t_j is $O(\log k)l_j$.*

Proof. Fix any j and suppose $t_j \in T_i$. Originally, the expected number of nodes mapped to t_j is $\sum_u \gamma(u, t_j) \leq l_j$. Let S_h be the change in this expectation of the number of nodes mapped to label t_j after deciding to which set T_i to map C_h. So $S_h = 1_{[C_h \text{ is mapped to } T_i]} \frac{\sum_{u \in C_h} \gamma(u,t_j)}{\gamma(C_h,T_i)}$. Using the Conditional Expectation Lemma, S_h is confined to $[0, l_j]$. So in particular $\frac{S_h}{l_j}$ is confined to $[0, 1]$ and we can then apply Lemma 5.7. Also note that $\sum_h E[S_h] = \sum_u \gamma(u, t_j)$. So with high probability $\sum_h S_h \leq O(\log k)l_j$. ∎

Lemma 6.5. *After the second-level rounding, with high probability for any j, the number of nodes mapped to label t_j is $O(\log k)l_j$.*

Proof. Fix any label t_j, and let R be the expected number of nodes assigned to t_j conditioned on the decisions of the first-level rounding procedure. We know from Lemma 6.4 that with high probability R is $O(\log k)l_j$. Let $u \in T_i$. Suppose that $f_1(C_h) = T_i$. For each $t_j \in T_i$, for each $u \in C_h$, define an indicator variable S_u which is one if u is assigned to t_j by the second-level rounding procedure.

Note that in applying Lemma 5.7, if $\delta > 0$ then

$$\Big(\frac{e^\delta}{(1+\delta)^{1+\delta}}\Big)^\mu \leq \Big(\frac{e}{1+\delta}\Big)^{\mu\delta}.$$

So this implies that with high probability $\sum_i S_i$ is $O(\log k)l_j$ i.e., when the mean R is already large ($O(\log k)l_j$), we can choose δ to be small and this will imply that the sum will not be much larger than its expectation R. So with high probability the number of nodes mapped to t_j is $O(\log k)l_j$. ∎

Theorem 1.4. There is a $O(\log k)$-approximation algorithm for determining the congestion C of an instance of Capacitated Metric Labeling.

Proof. This theorem follows from the fact that we check feasibility for the above linear program for all n possible values of C. The smallest value of C that will be feasible will be at most the optimal congestion C^*.

Given such a solution we can perform the **Two-Level Rounding Procedure** to get a solution in which the number of nodes assigned to any particular label is $O(\log k)C^* l_j$. ∎

7 Open Question

The main open question remaining in understanding Capacitated Metric Labeling is to determine if there is a polynomial-time bicriteria approximation algorithm that achieves a polylogarithmic approximation for the cost and multiplicatively violates the capacities of the labels by a $O(\log k)$ factor. One can draw an analogy with the question of scheduling unrelated parallel machines and of routing flow unsplittably. The classic paper of Lenstra, Shmoys and Tardos [23] gives a 2-approximation algorithm for the problem of scheduling unrelated parallel machines for the single objective of minimizing makespan. No polynomial-time approximation algorithm can achieve better than a $\frac{3}{2}$-approximation ratio for this question. Subsequently this question was generalized to an assignment problem in which each job incurs unrelated costs when scheduled on different machines. Each machine is given a time bound T_i, and the goal is to minimize the cost among all feasible solutions. Again, due to the hardness results in [23], no polynomial-time approximation algorithm can return a valid schedule if $P \neq NP$. This problem is known as the Generalized Assignment Problem. However, Shmoys and Tardos [28] were able to give a polynomial-time approximation algorithm that returns a solution of cost at most the cost of the optimal (valid) solution, except that each machine is additionally assigned at most $2T_i$ units of work. This is a generalization of the results in [23] because this approximation algorithm achieves a 2-approximation for the problem of minimizing makespan, but additionally achieves a 1-approximation for the cost.

Similarly, for Capacitated Metric Labeling we know that no polynomial-time approximation algorithm can achieve $O((\log k)^{1/2-\epsilon})$ ratio for the problem of minimizing the multiplicative violation of capacities so that there is a zero-cost solution. Yet for this problem, we demonstrated that there is a polynomial-time, $O(\log k)$-approximation algorithm for the problem of minimizing the multiplicative violation so that there is a zero-cost solution. So the natural open question is this: is there a bicriteria approximation algorithm for Capacitated Metric Labeling that achieves a polylogarithmic approximation in the cost, and simultaneously a $O(\log k)$ (or better) multiplicative violation of capacities? Perhaps there is a generalization of our two-stage rounding procedure (for approximating congestion) to one that also achieves a reasonable approximation for the cost function in the same way that [28] provides a generalization to [23].

Open Question 1. *Is there a polynomial-time bicriteria approximation algorithm for* Capacitated Metric Labeling *that achieves a polylogarithmic approximation ratio for the cost and a* $O(\log k)$ *multiplicative violation of the label capacities?*

An interesting note is that scheduling to minimize makespan on parallel unrelated machines is a special case of the single-source unsplittable flow problem in which the graph is a source and sink connected to a bipartite graph. Dinitz, Garg, and Goemans [12] (building on the work of Kleinberg in [20] and [19]) gave a substantial generalization of the result of [23] that achieves a similar 2 approximation in the violation of edge capacities in the single-source unsplittable flow problem for a general directed graph. Yet for this problem, no analogue of [28] is known either. If the problem is also equipped with costs, then no approximation algorithm is known that achieves a reasonable approximation in the costs and yet still violates the capacities by at most a factor of 2.

8 Acknowledgements

We would like to thank the anonymous reviewers for many helpful comments.

References

[1] K. Andreev and H. Räcke, *Balanced graph partitioning*, Theory of Computing Systems (2000), pp. 929-939.

[2] M. Andrews, J. Chuzhoy, V. Guruswami, S. Khanna, K. Talwar, and L. Zhang, *Inapproximability of edge-disjoint paths and low congestion routing on undirected graphs*, Electronic Colloquium on Computational Complexity (2007).

[3] M. Andrews, J. Chuzhoy, S. Khanna, and L. Zhang, *Hardness of the undirected edge-disjoint paths problem with congestion*, Foundations of Computer Science (2005), pp. 226-244.

[4] M. Andrews and L. Zhang, *Wavelength assignment in optical networks with fixed fiber capacity*, International Colloquium on Automata, Languages and Programming (2004), pp. 134-145.

[5] M. Andrews and L. Zhang, *Hardness of undirected edge-disjoint paths problem*, Symposium on Theory of Computing (2005), pp. 276-283.

[6] A. Archer, J. Fakcharoenphol, C. Harrelson, R. Krauthgamer, K. Talwar, and E. Tardos, *Approximate classification via earthmover metrics*, Symposium on Discrete Algorithms (2004), pp. 1072-1080.

[7] Y. Bartal, *Probabilistic approximation of metric spaces and its algorithmic applications*, Foundations of Computer Science (1996), pp. 184-193.

[8] G. Calinescu, H. Karloff, and Y. Rabani. *Approximation algorithms for the* 0*-extension problem*, SIAM Journal on Computing, 34 (2004), pp. 358-372.

[9] C. Chekuri, S. Khanna, J. Naor, and L. Zosin. *Approximation algorithms for the metric labeling problem via a new linear programming formulation*, Symposium on Discrete Algorithms (2001), pp. 109-118.

[10] J. Chuzhoy, V. Guruswami, S. Khanna, and K. Talwar, *Hardness of routing with congestion in directed graphs*, Symposium on Theory of Computing (2007), pp. 165-178.

[11] J. Chuzhoy and J. Naor, *The hardness of metric labeling*, Foundations of Computer Science (2004), pp. 108-114.

[12] Y. Dinitz, N. Garg, and M. Goemans, *On the single-source unsplittable flow problem*, Combinatorica (1999), pp. 17-41.

[13] G. Even, J. Naor, S. Rao, and B. Schieber, *Divide-and-conquer approximation algorithms via spreading metrics*, Journal of the ACM (2000), pp. 585-616.

[14] J. Fakcharoenphol, C. Harrelson, S. Rao, and K. Talwar, *An improved approximation algorithm for the* 0*-extension problem*, Symposium on Discrete Algorithms (2003), pp. 257-265.

[15] J. Fakcharoenphol, S. Rao, and K. Talwar, *A tight bound on approximation arbitrary metric by tree metrics*, Symposium on Theory of Computing (2003), pp. 448-455.

[16] U. Feige and R. Krauthgamer, *A polylogarithmic approximation of the minimum bisection*, SIAM Journal on Computing, 31 (2002), pp. 1090-1118.

[17] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman (1979).

[18] A. Gupta and E. Tardos, *Constant factor approximation algorithms for a class of classification problems*, Symposium on Theory of Computing (2000), pp. 652-658.

[19] J. Kleinberg, *Approximation algorithms for disjoint paths problems*, PhD dissertation, MIT (1996).

[20] J. Kleinberg, *Single-source unspittable flow*, Foundations of Computer Science (1996), pp. 68-77.

[21] J. Kleinberg and E. Tardos, *Approximation algorithms for classification problems with pairwise relationships: metric labeling and Markov random fields*, Journal of the ACM (2002), pp. 616-639.

[22] T. Leighton and S. Rao, *Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms*, Journal of the ACM (1999), pp. 787-832.

[23] J. Lenstra, D. Shmoys, and E. Tardos, *Approximation algorithms for scheduling unrelated parallel machines*, Mathematical Programming (1990), pp. 259-271.

[24] J. Naor and R. Schwartz, *Balanced metric labeling*, Symposium on Theory of Computing (2005), pp. 582-591.

[25] H. Räcke, *Minimizing congestion in general networks*, Foundations of Computer Science (2003), pp. 43-52.

[26] H. Räcke, *Optimal hierarchical decompositions for congestion minimization in networks*, Symposium on Theory of Computing (2008), pp. 255-264.

[27] J. Schmidt, A. Siegel, and A. Srinivasan, *Chernoff-Hoeffding bounds for applications with limited dependence*, SIAM Journal on Discrete Math (1995), pp. 223-250.

[28] D. Shmoys and E. Tardos, *An approximation algorithm for the generalized assignment problem*, Mathematical Programming (1993), pp. 461-474.