Simplex Partitioning via Exponential Clocks and the Multiway Cut Problem

[Extended Abstract]

Niv Buchbinder*
Tel Aviv University
niv.buchbinder@gmail.com

Joseph (Seffi) Naor*
Technion
naor@cs.technion.ac.il

Roy Schwartz
Microsoft Research
roysch@microsoft.com

ABSTRACT

The Multiway-Cut problem is a fundamental graph partitioning problem in which the objective is to find a minimum weight set of edges disconnecting a given set of special vertices called terminals. This problem is NP-hard and there is a well known geometric relaxation in which the graph is embedded into a high dimensional simplex. Rounding a solution to the geometric relaxation is equivalent to partitioning the simplex. We present a novel simplex partitioning algorithm which is based on *competing exponential clocks* and *distortion*. Unlike previous methods, it utilizes cuts that are not parallel to the faces of the simplex. Applying this partitioning algorithm to the multiway cut problem, we obtain a simple (4/3)-approximation algorithm, thus, improving upon the current best known result. This bound is further pushed to obtain an approximation factor of 1.32388. It is known that under the assumption of the unique games conjecture, the best possible approximation for the Multiway-Cut problem can be attained via the geometric relaxation.

Categories and Subject Descriptors

H.2 [**Theory of Computation**]: ANALYSIS OF ALGORITHMS AND PROBLEM COMPLEXITY

General Terms

Algorithms, Theory

Keywords

Approximation, Multiway, Cut, Randomized, Simplex

1. INTRODUCTION

Geometric embedding has been at the heart of approximation algorithms for NP-hard graph problems in recent years [2, 3, 5, 10, 17, 23, 24]. The Multiway-Cut problem in undirected graphs is a prime example for the success of this approach. In this problem we are given an edge weighted undirected graph $G = (V, E)$, $w : E \to \mathcal{R}^+$, and a set $T = \{t_1, t_2, \ldots, t_k\} \subseteq V$ of k terminals. The goal is to find a minimum weight subset of edges $X \subseteq E$ such that all terminals are disconnected in $(V, E \setminus X)$. The Multiway-Cut problem is known to be NP-hard even for $k = 3$ [9][1]. The first approximation algorithm for the Multiway-Cut problem was given by Dalhaus *et al.* [9]. It is a simple combinatorial heuristic: for each terminal t_i, compute a minimum cut separating it from all other terminals. Dalhaus *et al.* [9] showed that the union of the $k-1$ cheapest cuts (of the k cuts computed) is a $2(1-1/k)$-approximation. Note that the union of any $k-1$ cuts defines a feasible solution.

Călinescu *et al.* [5] suggested a geometric relaxation for the Multiway-Cut problem. In this relaxation each vertex $u \in V$ is embedded into the k-dimensional simplex $\Delta_k = \{x \in \mathcal{R}^k : x \geq 0,\ \sum_i x_i = 1\}$, while the terminals are mapped bijectively to the vertices of Δ_k. Călinescu *et al.* [5] observed that any rounding of this relaxation is in fact a partitioning of the simplex into k parts, one for each terminal (or equivalently a vertex of Δ_k). They presented a partitioning algorithm achieving an approximation guarantee of $3/2 - 1/k$. Building upon this result, Karger *et al.* [18] managed to obtain an improved guarantee of $1.3438 - \varepsilon_k$, where ε_k is a decreasing function of k that tends to 0 as k increases.

Conversely, it was shown by Freund and Karloff [13] that the geometric relaxation of Călinescu *et al.* [5] has an integrality gap of $8/(7 + 1/(k-1))$. Furthermore, Manokaran *et al.* [25] proved that the integrality gap of this relaxation can be translated to a hardness result for the Multiway-Cut problem with the exact same value, assuming the unique games conjecture. Hence, this result suggests that the best possible approximation guarantee for the Multiway-Cut problem can be obtained by rounding the geometric relaxation of Călinescu *et al.* [5]. Finally, Dahlhaus *et al.* [9] showed that Multiway-Cut is APX-hard. Thus, there exists a constant $c > 1$ such at no polynomial-time algorithm can find a solution within a factor of c of the optimum, unless P=NP.

We also note that some of the above mentioned techniques, developed in the context of Multiway-Cut, have found additional applications. The main idea of the algorithm of Călinescu *et al.* [5] is to iterate over the terminals in a random order and cut an ℓ_1 sphere of random radius around each terminal, thus partitioning the simplex. This idea was extended to general metrics, providing improved approxima-

*Supported by ISF grant 954/11 and BSF grant 2010426.

[1]For $k = 2$ it is simply the minimum $\{s, t\}$-cut problem in undirected graphs.

tions for the 0-Extension problem [4, 11] (whose study was originated by Karzanov [20]) and the probabilistic approximation of metrics by tree metrics [12].

1.1 Our Results

We present a new method for partitioning the simplex which is based on competing exponential clocks and on distorting the simplex. Unlike previous approximation algorithms for the Multiway-Cut problem [5, 18], our method generates cuts in the simplex that are not parallel to the faces of the simplex. We obtain simple approximation algorithms for the Multiway-Cut problem, as well as improved approximation guarantees. We believe that our new simplex partitioning method is of independent interest.

First, we present a simple (4/3)-approximation algorithm, as summarized in the following theorem.

THEOREM 1. *There exists a* $\left(\frac{4}{3} - \frac{4}{9k-6}\right)$*-approximation algorithm for the Multiway-Cut problem.*

Note that the above theorem already improves on the current best known approximation guarantee of [18]. We then manage to push the approximation guarantee further by a more careful use of our simplex partitioning method, as stated in the following theorem.

THEOREM 2. *There exists a* $\left(1.32388 - \frac{1}{2k}\right)$*-approximation algorithm for the Multiway-Cut problem.*

This theorem demonstrates that an approximation factor strictly better than 4/3 can be achieved.

Additionally, we show that the exponential clocks ingredient of our simplex partitioning algorithm can also be applied to the Uniform-Metric-Labeling problem, yielding a 2-approximation algorithm which is different from the known 2-approximation algorithm of Kleinberg and Tardos [22]. Surprisingly, even though both algorithms are different, we show that the algorithm of [22] can replace the exponential clocks ingredient of our simplex partitioning algoritm, resulting in the exact same guarantees as in Theorems 1 and 2 for the Multiway-Cut problem.

Finally, we note that competing exponential clocks can be used to round fractional solutions of more general covering problems[2]. We refer the reader to Appendix A for details.

1.2 Techniques

The improved approximation for Multiway-Cut uses two main separate ingredients which are then joined probabilistically (see Section 5). The first ingredient is a partitioning algorithm that uses a single *randomly chosen* point in the simplex to generate a partition, as opposed to previous works [5, 18] that use ℓ_1 *spheres* to partition the simplex. Let us provide a geometric description of this partition and for simplicity restrict the discussion to Δ_3. Given a point $Z \in \Delta_3$, it is used to define the boundaries between the different parts, thus, inducing a partitioning. This is shown in Figure 1. There are lines going from each terminal through Z to the opposite edge in Δ_3. For example, the line starting at terminal t_1 passes through Z and ends at point a where it hits the edge opposite to t_1. For each such line, the segment that lies between the terminal and Z is ignored. The ignored segments are represented as dashed, and the boundaries of the partition induced by Z are the remaining segments represented as solid. For example, the part that is assigned to t_1 in the partitioning induced by Z is the area defined by the points t_1, c, Z, and b.

The above partitioning method easily extends to any high dimensional simplex. It is important to note that this natural way for partitioning the simplex does not use face-parallel cuts, unlike [5, 18]. Our algorithm applies the above partitioning method by choosing a point Z uniformly at random using competing exponential clocks. The use of exponential clocks allows for a simple analysis of the probability of separating adjacent vertices between different parts.

The second ingredient used is a *distortion* of the simplex. This distortion is applied prior to partitioning the simplex, resulting in both increase and decrease of some of the distances. Given $u \in \Delta_k$, each of the first $k-1$ coordinates in u is decreased by applying a transformation to it, while the value of the kth coordinate is chosen so that u is mapped back to the simplex. For example, Figure 1 contains the quadratic distortion of Δ_3, $(u_1, u_2, u_3) \to \left(u_1^2, u_2^2, 1 - u_1^2 - u_2^2\right)$, used by our algorithm. The boundary of Δ_3 is dashed while the boundary of the distorted Δ_3 appears as solid.

1.3 Related Work

In addition to the works already mentioned, several other special cases and variants of Multiway-Cut were considered in the literature. Cunningham and Teng [8] and Karger *et al.* [18] present a tight approximation factor of 12/11 when $k = 3$. For $k = 4, 5$, [18] provide approximation factors of 1.1539 and 1.2161, respectively. For dense unweighted graphs, Arora *et al.* [1] and Frieze and Kannan [14] provide a polynomial time approximation scheme. The Node-Multiway-Cut problem asks for the least weight subset of vertices whose removal from the graph disconnects all terminals. This variant was studied by Garg *et al.* [16] who present a $2(1 - 1/k)$-approximation algorithm for the problem. They also prove that any improvement to the latter factor would also lead to an improvement of the approximation guarantee for Vertex-Cover, for which it is known that no approximation better than 2 can be achieved assuming the unique games conjecture (Khot and Regev [21]). The Directed-Multiway-Cut problem asks for the least weight subset of edges whose removal from the graph disconnects all directed paths connecting terminals. Clearly, Directed-Multiway-Cut generalizes Node-Multiway-Cut. For Directed-Multiway-Cut, Naor and Zosin [26] give a 2-approximation algorithm, improving upon the $O(\log k)$-approximation of Garg *et al.* [16]. The Multicut problem resembles Multiway-Cut, however, its goal is to separate k pairs of terminals $\{s_i, t_i\}$. The best known approximation for Multicut is $O(\log k)$ and is given by Garg *et al.* [15].

Several additional problems are closely related to Multiway-Cut, and two prominent examples are 0-Extension and Metric-Labeling. For the 0-Extension problem, Călinescu *et al.* [4] provide an approximation of $O(\log k)$, which was later improved by Fakcharoenphol *et al.* [11] to $O\left(\log k / \log\log k\right)$. Both results are obtained by rounding the *metric completion* relaxation. For the Metric-Labeling problem, Kleinberg and Tardos [22] provide an approximation guarantee

[2] Multiway-Cut can be viewed as a covering problem in which the goal is to cover the vertices of the graph by k sets, each containing exactly one terminal. If some of the sets intersect, uncrossing can be used to turn the cover into a multiway partitioning without increasing the cost of the solution.

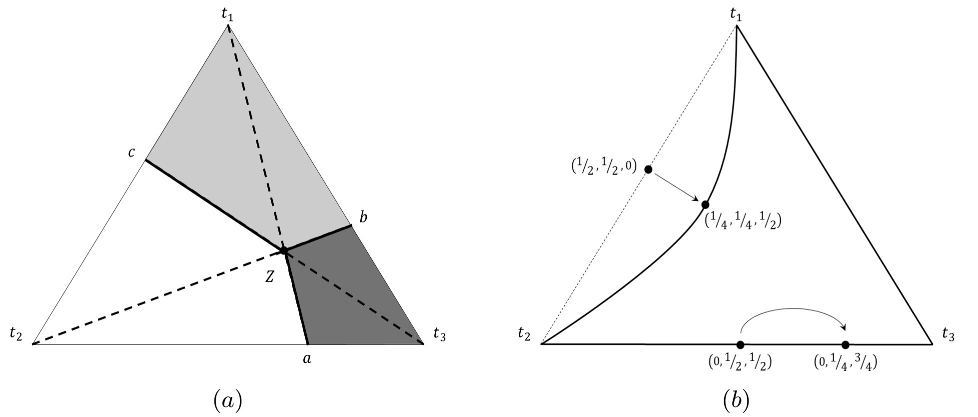

Figure 1: (a) The partition of Δ_3 induced by Z. (b) Quadratic distortion of Δ_3.

of $O(\log k)$ by building upon the tight probabilistic approximation of metrics by tree metrics [12]. Chekuri *et al.* [6] extend the geometric relaxation of Călinescu *et al.* [5] to the more general Metric-Labeling problem using *earthmover metrics*, thus yielding a unified treatment of the problem for various metrics. For the geometric relaxation with the earthmover metric, Karloff *et al.* [19] prove an integrality gap of $\Omega(\log k)$ for Metric-Labeling and $\Omega(\sqrt{\log k})$ for 0-Extension. Both integrality gaps translate to hardness results with the same values assuming the unique games conjecture [25]. Without this assumption, the corresponding hardness results are $\Omega(\sqrt{\log k})$ for Metric-Labeling [7] and $\Omega(\log^{1/4} k)$ for 0-Extension [19].

2. PRELIMINARIES

An instance of the Multiway-Cut problem consists of an edge weighted undirected graph $G = (V, E)$, $w : E \to \mathbb{R}^+$, and a set of terminals $T = \{t_1, t_2, \ldots, t_k\} \subseteq V$. The goal is to find the least weight subset of edges $X \subseteq E$ such that all terminals are disconnected in $(V, E \setminus X)$. Equivalently, the output is an assignment $\ell : V \to T$ with the property that $\ell(t_i) = t_i$, $\forall i = 1, \ldots, k$. For an edge $e = (u, v) \in E$, $e \in X$ if and only if $\ell(u) \neq \ell(v)$.

2.1 The Simplex Relaxation for Multiway Cut

We use the simplex relaxation first proposed by [5]. Consider the k-dimensional simplex $\Delta_k = \{x \in \mathcal{R}^k : x \geq 0,\ \sum_i x_i = 1\}$. Denote by e_i the ith vertex of Δ_k, i.e., the standard basis vector which has a 1 in the ith coordinate and 0 elsewhere. The relaxation embeds every vertex $u \in V$ into Δ_k and each terminal $t_i \in T$ is embedded to e_i. Intuitively, every point in Δ_k corresponds to a distribution over the set of terminals T. For simplicity of presentation, we abuse notation and use u for both the vertex and its embedding in the simplex.

$$\min \quad \frac{1}{2} \sum_{e=(u,v)\in E} w(e) \cdot ||u - v||_1 \qquad (1)$$

$$\text{s.t.} \quad u \in \Delta_k \qquad \forall u \in V \qquad (2)$$

$$t_i = e_i \qquad \forall i = 1, 2, \ldots, k \qquad (3)$$

It is known [5] that the above relaxation can be rewritten as a linear program, thus an optimal fractional solution can be efficiently computed.

We note that any integral solution $\ell : V \to T$ to the Multiway-Cut problem defines a feasible embedding in the simplex of the same value. To see that, for every $u \in V$, map u to $e_i \in \Delta_k$, if $\ell(u) = t_i$. Hence, the simplex relaxation is indeed a lower bound on the value of an optimal solution.

For simplicity, as was noted by [5], when bounding the separation probability of two points $u, v \in \Delta_k$, one can assume without loss of generality that u and v differ in only two coordinates. The reason for this is that an edge $(u, v) \in E$ can be subdivided in such a way that this property holds and the cost remains unchanged, since the contribution of the edge to (1) is proportional to $\|u - v\|_1$. We refer the reader to [5] for the exact details. Thus, in the sequel we always assume that u and v differ in coordinates 1 and 2.

2.2 Exponential Clocks

By *exponential clocks* we mean competing independent exponential random variables. An exponential clock wins a competition if it has the smallest value among all participating exponential clocks. A random variable X is distributed according to the exponential distribution with rate λ if it has density $f_X(x) = \lambda e^{-\lambda x}$ for every $x \geq 0$ and $f_X(x) = 0$ otherwise. We denote this by $X \sim \exp(\lambda)$. Additionally, we use the following well known properties of the exponential distribution:

1. If $X \sim \exp(\lambda)$ and $c > 0$, then $\frac{X}{c} \sim \exp(\lambda c)$.
2. Let $X_1, \ldots, X_k$ be independent random variables with $X_i \sim \exp(\lambda_i)$:
   (a) $\min\{X_1, \ldots, X_k\} \sim \exp(\lambda_1 + \ldots + \lambda_k)$.
   (b) $\Pr[X_i \leq \min_{j\neq i}\{X_j\}] = \frac{\lambda_i}{\lambda_1+\ldots+\lambda_k}$.

3. ALGORITHM I - COMPETING EXPONENTIAL CLOCKS

In this section we present a new method for partitioning the simplex which is based on competing exponential clocks. Intuitively, an independent exponential clock with rate $\lambda = 1$ is associated with each vertex of Δ_k, one for each terminal in T. Given a fractional point $u \in \Delta_k$, it is assigned to a terminal as follows. The ith clock (of terminal t_i) is scaled by u_i and thus its rate becomes u_i (by property (1) of the exponential distribution). Then, the k scaled clocks compete and u is assigned to the terminal corresponding to the clock winning the competition (i.e., the smallest clock).

A precise description of the algorithm is given in Algorithm 1. The algorithm receives as input an optimal solution to the simplex relaxation, $\{u\}_{u\in V} \subseteq \Delta_k$, and outputs the assignment $\ell : V \to T$, assigning each vertex to a terminal in T.

Algorithm 1: Exponential-Clocks($\{u\}_{u\in V} \subseteq \Delta_k$)

1 choose i.i.d random variables $Z_i \sim \exp(1)$, $i = 1, \ldots, k$.
2 $\forall u \in V$: $\ell(u) \leftarrow \text{argmin}\left\{\frac{Z_i}{u_i} \ : \ i = 1, 2, \ldots, k\right\}$.

It is easy to verify that for each terminal $t_i \in T$, $\ell(t_i) = t_i$. This happens since $(t_i)_j = 0$ for every $j \neq i$ and $(t_i)_i = 1$. Thus, with probability that equals 1, $\ell(t_i) = t_i$, and Algorithm 1 partitions the simplex in a way that produces a feasible integral solution for Multiway-Cut.

3.1 Geometric Interpretation

We now demonstrate the connection between the exponential clocks used in Algorithm 1 and the claimed geometric partitioning of the simplex from Section 1.2 and Figure 1. Denote by $Z = (Z_1, Z_2, \ldots, Z_k)$ the vector of exponential clocks chosen by Algorithm 1.

First, it is known (for example, see [27]) that $Z/\left(\sum_{i=1}^{k} Z_i\right)$ is a uniformly chosen random point in Δ_k. Second, note that Algorithm 1 is indifferent to any normalization of the vector Z. Hence, one can assume without loss of generality that the vector of exponential clocks Z chosen by Algorithm 1 is indeed a uniformly chosen random point in Δ_k.

As in Section 1.2 and Figure 1, let us restrict the discussion to Δ_3. A fractional point $u = (u_1, u_2, u_3) \in \Delta_3$ is assigned by Algorithm 1 to the winning clock (i.e., the smallest) among the scaled exponential clocks Z_1/u_1, Z_2/u_2, and Z_3/u_3. The only point in the simplex for which all the clocks are equal is Z itself. Next, consider all points in the simplex for which exactly two of the clocks are equal. Focus, for example, on the clocks of terminals t_2 and t_3. All such points must satisfy the linear equation $Z_2/u_2 = Z_3/u_3$, whose solution forms a straight line that starts at t_1, passes through Z, and ends at the edge opposite to t_1 in Δ_3. As in Figure 1, this happens at point

$$a = (0, Z_2/(Z_2 + Z_3), Z_3/(Z_2 + Z_3)).$$

The dashed segment of this line, connecting t_1 and Z, corresponds to all points $u \in \Delta_3$ for which the clock of terminal t_1 wins over the clocks of terminals t_2 and t_3, i.e., $Z_1/u_1 < Z_2/u_2 = Z_3/u_3$. The solid segment of this line, connecting Z and a, corresponds to all points $u \in \Delta_3$ for which the clocks of terminals t_2 and t_3 both win over the clock of terminal t_1, i.e., $Z_1/u_1 > Z_2/u_2 = Z_3/u_3$. Therefore, one can conclude that this solid segment, as claimed in Section 1.2, is indeed the boundary between the part assigned to t_2 and the part assigned to t_3. Repeating the argument for all terminal pairs in Δ_3 yields that Algorithm 1 indeed generates the geometric partitioning of Δ_3 as shown in Figure 1.

3.2 Analysis

We focus on a particular edge $(u, v) \in E$ and bound the separation probability of u and v. As mentioned in Section 2, we can assume without loss of generality that u and v differ in only two coordinates.

LEMMA 1. *Let*

$$u = (u_1, u_2, u_3, \ldots, u_k) \text{ and}$$
$$v = (u_1 + \varepsilon, u_2 - \varepsilon, u_3, \ldots, u_k),$$

where $u, v \in \Delta_k$ for some $\varepsilon > 0$. Then,

$$\Pr[\ell(u) \neq \ell(v)] \leq \varepsilon(2 - u_1 - u_2).$$

PROOF. For each $1 \leq i \leq k$, let A_i be the event that $\ell(u) = \ell(v) = i$. Let us calculate $\Pr[A_i]$, for all i, starting with $\Pr[A_1]$.

$$\begin{aligned}\Pr[A_1] &= \Pr\left[\frac{Z_1}{u_1} \leq \min\left\{\frac{Z_2}{u_2}, \ldots, \frac{Z_k}{u_k}\right\},\right.\\ &\qquad \left.\frac{Z_1}{u_1+\varepsilon} \leq \min\left\{\frac{Z_2}{u_2-\varepsilon}, \frac{Z_3}{u_3}, \ldots, \frac{Z_k}{u_k}\right\}\right]\\ &= \Pr\left[\frac{Z_1}{u_1} \leq \min\left\{\frac{Z_2}{u_2}, \ldots, \frac{Z_k}{u_k}\right\}\right]\\ &= \frac{u_1}{u_1 + u_2 + \ldots + u_k} = u_1.\end{aligned}$$

The one before last equality follows from the fact that the Z_i-s are independent, and from properties (1) and (2b) of the exponential distribution (see Section 2). Let us calculate $\Pr[A_2]$.

$$\begin{aligned}\Pr[A_2] &= \Pr\left[\frac{Z_2}{u_2} \leq \min\left\{\frac{Z_1}{u_1}, \frac{Z_3}{u_3}, \ldots, \frac{Z_k}{u_k}\right\},\right.\\ &\qquad \left.\frac{Z_2}{u_2-\varepsilon} \leq \min\left\{\frac{Z_1}{u_1+\varepsilon}, \frac{Z_3}{u_3}, \ldots, \frac{Z_k}{u_k}\right\}\right]\\ &= \Pr\left[\frac{Z_2}{u_2} \leq \frac{u_2-\varepsilon}{u_2}\min\left\{\frac{Z_1}{u_1+\varepsilon}, \frac{Z_3}{u_3}, \ldots, \frac{Z_k}{u_k}\right\}\right]\\ &= \frac{u_2}{\frac{u_2}{u_2-\varepsilon}\cdot(u_1 + \varepsilon + u_3 + \ldots + u_k) + u_2} = u_2 - \varepsilon.\end{aligned}$$

The one before last equality follows from the fact that the Z_i-s are independent, and from properties (1) and (2b) of the exponential distribution (see Section 2). Finally, let us calculate $\Pr[A_i]$ for all $i = 3, \ldots, k$.

$$\begin{aligned}\Pr[A_i] &= \Pr\left[\frac{Z_i}{u_i} \leq \min\left\{\frac{Z_1}{u_1}, \frac{Z_2}{u_2}, \frac{Z_3}{u_3}, \ldots, \frac{Z_k}{u_k}\right\},\right.\\ &\qquad \left.\frac{Z_i}{u_i} \leq \min\left\{\frac{Z_1}{u_1+\varepsilon}, \frac{Z_2}{u_2-\varepsilon}, \frac{Z_3}{u_3}, \ldots, \frac{Z_k}{u_k}\right\}\right]\\ &= \Pr\left[\frac{Z_i}{u_i} \leq \min\left\{\frac{Z_1}{u_1+\varepsilon}, \frac{Z_2}{u_2}, \frac{Z_3}{u_3}, \ldots, \frac{Z_k}{u_k}\right\}\right]\\ &= \frac{u_i}{u_1 + \varepsilon + u_2 + u_3 + \ldots + u_k} = \frac{u_i}{1+\varepsilon}.\end{aligned}$$

Again, the one before last equality follows from the fact that all the Z_i-s are independent, and from properties (1) and (2b) of the exponential distribution (see Section 2). To conclude the proof, note that:

$$
\begin{aligned}
\Pr\left[\ell(u) \neq \ell(v)\right] &= 1 - \sum_{i=1}^{k} \Pr\left[A_i\right] \\
&= 1 - u_1 - (u_2 - \varepsilon) - \sum_{i=3}^{k} \frac{u_i}{1+\varepsilon} \\
&= \varepsilon \left(\frac{2 - u_1 - u_2 + \varepsilon}{1+\varepsilon} \right) \\
&\leq \varepsilon \left(2 - u_1 - u_2\right).
\end{aligned}
$$

The last inequality follows since $u_1 + u_2 \leq 1$. $\square$

4. ALGORITHM II - SIMPLEX DISTORTION

In this section we present a simple method for partitioning the simplex once it is distorted. Intuitively, we apply to each coordinate of the simplex a non-linear transformation, thus distorting the simplex and changing the ℓ_1 distances between points non-linearly. Once the simplex is distorted, we just choose a uniform random threshold and assign all vertices to terminals by going over the terminals in an arbitrary order[3]. Finally, all unassigned vertices are assigned to the last terminal in this order.

A precise description of the algorithm appears in Algorithm 2. The algorithm receives as input an optimal solution to the simplex relaxation, $\{u\}_{u\in V} \subseteq \Delta_k$, and outputs the assignment $\ell : V \to T$, assigning each vertex to a terminal in T. We choose the quadratic transformation (as can be seen in line 3 of the algorithms) and order the terminals according to their index (as can be seen in line 2 of the algorithm).

Algorithm 2: Distorting-Simplex($\{u\}_{u\in V} \subseteq \Delta_k$)

```
1  choose r ~ unif[0,1].
2  for i=1 to k − 1
3      ∀u ∈ V s.t. u is unassigned and r ≤ u_i^2: ℓ(u) ← i.
4  for all unassigned u ∈ V, ℓ(u) ← k.
```

As in Algorithm 1, it is easy to verify that for each terminal t_i, $\ell(t_i) = t_i$. This is true since $(t_i)_j^2 = 0$ for every $j \neq i$ and $(t_i)_i^2 = 1$. Thus, with probability that equals 1, $\ell(t_i) = t_i$, and Algorithm 2 partitions the simplex in a way that produces a feasible integral solution for Multiway-Cut.

Remark 1. *Instead of distorting the simplex, one can equivalently define Algorithm 2 by choosing the threshold r non-uniformly. Specifically, r is distributed like $\sqrt{\theta}$ where $\theta \sim unif[0,1]$. This distribution favors values closer to 1 over values closer to 0. Thus, Algorithm 2 is more likely to cut close to the terminals. We note that this way of defining Algorithm 2, though having a simple probabilistic interpretation, loses the geometric insight.*

4.1 Geometric Interpretation

Algorithm 2 applies the following transformation, distorting Δ_k prior to partitioning the simplex via face-parallel cuts:

$$
(u_1, \ldots, u_{k-1}, u_k) \ \to \left(u_1^2, \ldots, u_{k-1}^2, 1 - \sum_{i=1}^{k-1} u_i^2 \right).
$$

Consider two vertices $u, v \in \Delta_k$ and assume that they differ in only two coordinates. Intuitively, the transformation decreases the ℓ_1 distance between u and v in the case that the two coordinates are small, and increases the ℓ_1 distance between u and v in the case that the two coordinates are large.

We note that an alternative way of distorting Δ_k, with respect to the separation probability guarantee of Lemma 2, is to apply the quadratic transformation to all coordinates and add the remaining value to a new coordinate (denoted by k+1). Formally, Δ_k can be distorted into Δ_{k+1} as follows:

$$
(u_1, \ldots, u_{k-1}, u_k) \ \to \left(u_1^2, \ldots, u_{k-1}^2, u_k^2, 1 - \sum_{i=1}^{k} u_i^2 \right).
$$

4.2 Analysis

We focus on a particular edge $(u, v) \in E$ and bound the separation probability of u and v. As mentioned in Section 2, we can assume without loss of generality that u and v differ in only two coordinates.

Lemma 2. *Let*

$$
\begin{aligned}
u &= (u_1, u_2, u_3, \ldots, u_k) \ \text{and} \\
v &= (u_1 + \varepsilon, u_2 - \varepsilon, u_3, \ldots, u_k),
\end{aligned}
$$

where $u, v \in \Delta_k$ for some $\varepsilon > 0$. Then,

$$
\Pr\left[\ell(u) \neq \ell(v)\right] \leq 2\epsilon(u_1 + u_2).
$$

Proof. Clearly, if u and v are not assigned in iterations 1 and 2, then $\ell(u) = \ell(v)$. Thus, a necessary condition for the event $\ell(u) \neq \ell(v)$ is that *at least* one of the following two happens:

1. In the beginning of iteration 1, both u and v are unassigned; then, u is not assigned to terminal t_1 and v is assigned to terminal t_1 (recall that $v_1 = u_1 + \varepsilon$).
2. In the beginning of iteration 2, both u and v are unassigned; then v is not assigned to terminal t_2 and u is assigned to terminal t_2 (recall that $v_2 = u_2 - \varepsilon$).

The former event happens only if $r \in [u_1^2, (u_1 + \varepsilon)^2]$, while the latter event happens only if $r \in [(u_2 - \varepsilon)^2, u_2^2]$. From the union bound on both events one can conclude that:

$$
\begin{aligned}
\Pr\left[\ell(u) \neq \ell(v)\right] &\leq \left((u_1 + \varepsilon)^2 - u_1^2\right) + \left(u_2^2 - (u_2 - \varepsilon)^2\right) \\
&= 2\varepsilon(u_1 + u_2),
\end{aligned}
$$

thus completing the proof. $\square$

Remark 2. *In contrast to Lemma 1, one* cannot *assume without loss of generality that the two coordinates in which u and v differ are 1 and 2. The reason is that in Algorithm 2 the coordinates are not symmetric: all unassigned vertices are assigned to terminal t_k in the last iteration. If both the two coordinates in which u and v differ are not the last coordinate, then clearly Lemma 2 holds. Otherwise, it is the case that one of the two coordinates in which u and v differ is the last coordinate. Obviously, in the latter case the separation probability of u and v can only decrease when compared to the bound of Lemma 2. Thus, one can always use the upper bound on the separation probability guaranteed by Lemma 2.*

[3]We note that our threshold cutting is similar to the one used by Călinescu *et al.* [5]. However, the threshold cutting proposed here is simpler as no random order over the terminals is required and each terminal can even choose a separate threshold independently.

5. AN IMPROVED APPROXIMATION FOR MULTIWAY CUT

In this section we show how to combine Algorithms 1 and 2 to obtain an improved approximation factor for the Multiway-Cut problem. The approximation factor is the ratio between the expected weight of our solution and the value of the objective function of the simplex relaxation, given in (1). By linearity of expectation we can bound this ratio separately for each edge of the graph. Thus, suppose that an upper bound on the separation probability of an edge $(u, v) \in E$ is available:

$$\Pr[\ell(u) \neq \ell(v)] \leq \beta\varepsilon,$$

where $\frac{1}{2}||u - v||_1 = \varepsilon$. Then, it implies a β-approximation for Multiway-Cut.

5.1 A $(\frac{4}{3})$-Approximation

Our analysis of Algorithms 1 and 2 (Lemmas 1 and 2), shows that both algorithms, when applied separately, yield only a 2-approximation for Multiway-Cut. What happens if we take a random combination of the two algorithms? Consider an edge (u, v). The behavior of the two algorithms is different with respect to the values of the two coordinates by which u and v differ. The bad case for Algorithm 1 is when the value of the two coordinates is small, whereas the bad case for Algorithm 2 is when the value of the two coordinates is large. Thus, intuitively, a random combination of both algorithms should yield an improved approximation factor, as given by Algorithm 3.

Algorithm 3: Combination($\{u\}_{u \in V} \subseteq \Delta_k$)

1 with probability 2/3 execute Algorithm 1.
2 otherwise (with probability 1/3) execute Algorithm 2.

Let us analyze the probability that an edge $(u, v) \in E$ is separated by Algorithm 3. Again, as mentioned in Section 2, one can assume without loss of generality that u and v differ in only two coordinates. Let $u = (u_1, u_2, u_3, \ldots, u_k)$ and $v = (u_1 + \varepsilon, u_2 - \varepsilon, u_3, \ldots, u_k)$ where $u, v \in \Delta_k$ for some $\varepsilon > 0$. Applying Lemmas 1 and 2, one can upper bound the probability that edge $(u, v) \in E$ is separated as follows:

$$\Pr[\ell(u) \neq \ell(v)] \leq \frac{2}{3} \cdot \varepsilon(2 - u_1 - u_2) + \frac{1}{3} \cdot 2\varepsilon(u_1 + u_2) = \frac{4}{3}\varepsilon.$$

Thus, Algorithm 3 is a (4/3)-approximation algorithm for Multiway-Cut, since the contribution of an edge $(u, v) \in E$ to the objective function (1) of the simplex relaxation is $\frac{1}{2}||u - v||_1 = \varepsilon$. However, the approximation factor can be slightly improved so as to yield Theorem 1.

Proof of Theorem 1. One can improve Algorithm 2 by either iterating over the terminals in random order (as in [5]), or simply choosing the last terminal in the order uniformly at random. If vertices u and v are unassigned in the last iteration, then they cannot be separated. The probability of each terminal to be last in the random order is $1/k$. Hence, this modification of Algorithm 2 changes the guarantee of Lemma 2 to

$$2\varepsilon(u_1 + u_2)\left(1 - \frac{1}{k}\right).$$

Plugging this improved guarantee and changing the probability of choosing Algorithm 1 from 2/3 to $\frac{2k-2}{3k-2}$ concludes the proof. $\square$

5.2 Beyond $(\frac{4}{3})$-Approximation

In this section we show how to push the approximation guarantee further by a careful distortion of the simplex, i.e., modifyinging Algorithm 2. The changes we apply to Algorithm 2 are the following:

1. Iterate over the terminals in a uniformly random order (line 2).
2. Change the threshold condition to $r \leq u_i^{\alpha}$ for $\alpha = 1.78061$ (line 3).

The improved algorithm is identical to Algorithm 3, except for two differences. First, it uses the modified Algorithm 2 instead of the original one. Second, it executes Algorithm 1 with probability $p = 0.604503$ and the modified Algorithm 2 with probability $1 - p = 0.395497$.

Proof of Theorem 2. Assume without loss of generality, as mentioned in Section 2, that u and v differ in only two coordinates. Let $u = (u_1, u_2, u_3, \ldots, u_k)$ and $v = (u_1 + \varepsilon, u_2 - \varepsilon, u_3, \ldots, u_k)$, where $u, v \in \Delta_k$ for some $\varepsilon > 0$. Additionally, one can assume without loss of generality that $\varepsilon \leq 1/(ck)$ for some small enough absolute constant c and that $u_2 - \varepsilon \geq u_1 + \varepsilon$. If this property does not hold then edge (u, v) can be subdivided into several edges satisfying the property. The analysis is then carried out for each edge separately.

Recall that by Lemma 1, the separation probability of Algorithm 1 is:

$$\Pr[\ell(u) \neq \ell(v)] \leq \varepsilon(2 - u_1 - u_2). \tag{4}$$

Building upon the proof of [5] when considering the modified Algorithm 2, the separation probability is:

$$\Pr[\ell(u) \neq \ell(v)] \leq \left(1 - \frac{1}{k}\right)\left(u_2^{\alpha} - (u_2 - \varepsilon)^{\alpha}\right) + \frac{1}{2}\left((u_1 + \varepsilon)^{\alpha} - u_1^{\alpha}\right). \tag{5}$$

The first summand on the right hand side of (5) corresponds to the event that $r \in [(u_2 - \varepsilon)^{\alpha}, u_2^{\alpha}]$ and terminal t_2 is not the last one in the random order. The second summand on the right hand side of (5) corresponds to the event that $r \in [u_1^{\alpha}, (u_1 + \varepsilon)^{\alpha}]$ and terminal t_1 precedes terminal t_2 in the random order. If terminal t_2 precedes terminal t_1 when $r \in [u_1^{\alpha}, (u_1 + \varepsilon)^{\alpha}]$, then u and v are not separated since $u_2 - \varepsilon \geq u_1 + \varepsilon$.

Choosing $\alpha = 1.78061$, the function u^{α} is convex and increasing for $u \in [0, 1]$, thus:

$$\left(1 - \frac{1}{k}\right)\left(u_2^{\alpha} - (u_2 - \varepsilon)^{\alpha}\right) + \frac{1}{2}\left((u_1 + \varepsilon)^{\alpha} - u_1^{\alpha}\right) \leq \varepsilon \cdot \alpha\left(\left(1 - \frac{1}{k}\right)u_2^{\alpha - 1} + \frac{1}{2}(u_1 + \varepsilon)^{\alpha - 1}\right). \tag{6}$$

Plugging Inequality (6) into Inequality (5) and combining the result with Inequality (4), one can get that the total separation probability satisfies:

$$\Pr[\ell(u) \neq \ell(v)] \leq \varepsilon \Bigg[p(2-u_1-u_2) + \alpha(1-p)\left(\left(1-\frac{1}{k}\right)u_2^{\alpha-1} + \frac{1}{2}(u_1+\varepsilon)^{\alpha-1}\right)\Bigg]. \quad (7)$$

Let us focus on the right hand side of (7). Defining $w = u_1 + u_2$ and exchanging variable u_1 for $u_1 = w - u_2$ yields the following function:

$$g_{\alpha,p}(u_2, w) \triangleq p(2-w) + \alpha(1-p)\left(\left(1-\frac{1}{k}\right)u_2^{\alpha-1} + \frac{1}{2}(w-u_2+\varepsilon)^{\alpha-1}\right), \quad (8)$$

where $0 \leq w \leq 1$ and $w/2+\varepsilon \leq u_2 \leq w$ (recall that $u_2 - \varepsilon \geq u_1 + \varepsilon$). Given any fixed $0 \leq w \leq 1$, $g_{\alpha,p}(u_2, w)$ is maximized when:

$$u_2 = C \cdot (w+\varepsilon), \qquad C = \frac{\left(1-\frac{1}{k}\right)^{\frac{1}{2-\alpha}}}{\left(\frac{1}{2}\right)^{\frac{1}{2-\alpha}} + \left(1-\frac{1}{k}\right)^{\frac{1}{2-\alpha}}}. \quad (9)$$

Equation (9) can be verified by solving the equation:

$$\frac{\partial g_{\alpha,p}(u_2,w)}{\partial u_2} = 0$$

and verifying that $\frac{\partial^2 g_{\alpha,p}(u_2,w)}{\partial u_2^2}$ is non-positive in the given range (recall that $1 < \alpha < 2$).

Next, given Equation (9), let us find the w that maximizes $g_{\alpha,p}(C\cdot(w+\varepsilon), w)$. The value of w that maximizes $g_{\alpha,p}(C\cdot(w+\varepsilon), w)$ is:

$$w = D^{\frac{1}{2-\alpha}} - \varepsilon, \qquad D = \frac{\alpha(\alpha-1)(1-p)\left[\left(1-\frac{1}{k}\right)C^{\alpha-1} + \frac{1}{2}(1-C)^{\alpha-1}\right]}{p}. \quad (10)$$

Again, Equation (10) can be verified by solving the equation:

$$\frac{\partial g_{\alpha,p}(C\cdot(w+\varepsilon),w)}{\partial w} = 0$$

and verifying that $\frac{\partial^2 g_{\alpha,p}(C\cdot(w+\varepsilon),w)}{\partial w^2}$ is non-positive in the given range (recall that $1 < \alpha < 2$ and $0 < p < 1$).

Finally, given the above values of u_2 and w in Equations (9) and (10) respectively, one can upper bound the right hand side of Inequality (7) as follows:

$$\varepsilon\Bigg\{p\left(2 - D^{\frac{1}{2-\alpha}} + \varepsilon\right) + \alpha(1-p)\cdot \left[\left(1-\frac{1}{k}\right)C^{\alpha-1}D^{\frac{\alpha-1}{2-\alpha}} + \frac{1}{2}\left(D^{\frac{1}{2-\alpha}} - CD^{\frac{1}{2-\alpha}}\right)^{\alpha-1}\right]\Bigg\}.$$

The latter bound can be rewritten as:

$$\Bigg(p\left(2 - D^{\frac{1}{2-\alpha}}\right) + \quad (11)$$

$$\alpha(1-p)\left[C^{\alpha-1}D^{\frac{\alpha-1}{2-\alpha}} + \frac{1}{2}\left(D^{\frac{1}{2-\alpha}} - CD^{\frac{1}{2-\alpha}}\right)^{\alpha-1}\right]\Bigg) -$$

$$\left(\alpha(1-p)C^{\alpha-1}D^{\frac{\alpha-1}{\alpha-2}} - \varepsilon k p\right)\frac{1}{k} \quad (12)$$

It can be verified that Summand (11) is an increasing function of k and is always upper bounded by 1.32388 when $\alpha = 1.78061$ and $p = 0.604503$. Additionally, Summand (12) is always at least $1/(2k)$ when $\varepsilon \leq 1/(ck)$, for some small enough absolute constant c, when $\alpha = 1.78061$ and $p = 0.604503$. This provides an overall guarantee of $1.32388 - 1/(2k)$, hence concluding the proof. □

Remark 3. *The dependance on k of the approximation guarantee of Theorem 2 can be improved. However, this improvement is most relevant for small values of k and does not improve the constant* 1.32388*. Since our main interest is the general case of* Multiway-Cut*, i.e. large k, this improved analysis is deferred to a full version of the paper.*

6. UNIFORM METRIC LABELING

In a typical classification problem, one wishes to assign labels to a set of objects so as to optimize some measure of the quality of the labeling. The Metric-Labeling problem, introduced by Kleinberg and Tardos [22], captures a broad range of classification problems in which the quality of a labeling depends on the pairwise relations between the underlying set of objects. The input to a Metric-Labeling instance is a an edge weighted undirected graph $G = (V, E)$, $w : E \to \mathcal{R}^+$, and a set of k labels L. The objective is to find a labeling, a function $f : V \to L$, that maps vertices to labels, where the cost of f, denoted by $Q(f)$, has two components.

- For each $u \in V$, there is a non-negative assignment cost $c(u, i)$ to label u with i. This cost reflects the relative likelihood of assigning labels to u.
- For each pair of objects u and v, the edge weight $w(u, v)$ measures the strength of their relationship. This is modeled in the objective function by the term $w(u, v)\cdot d(f(u), f(v))$ where $d(\cdot,\cdot)$ is a distance function on the labels L.

Thus,

$$Q(f) = \sum_{u\in V} c(u, f(u)) + \sum_{(u,v)\in E} w(u, v)\cdot d(f(u), f(v)),$$

and the goal is to find a labeling of minimum cost. In the Metric-Labeling problem, the distance function d is assumed to be a metric. In this section we focus on the Uniform-Metric-Labeling problem, the special case where d is the *uniform metric*, i.e., $d(i, j) = 1$ if $i \neq j$. Note that Multiway-Cut is a special case of Uniform-Metric-Labeling. The terminals correspond to the labels; for each terminal $t_i \in T$, the assignment cost of label i is 0 and the assignment cost of label $j \neq i$ is ∞. For all non-terminals the assignment cost is 0 for all labels. Kleinberg and Tardos [22] use the geometric relaxation of [5] for the Uniform-Metric-Labeling problem and obtain a 2-approximation for the problem[4]. This bound is tight as they present for every k an instance for which the integrality gap of the relaxation is $2(1 - 1/k)$.

A precise description of the Kleinberg-Tardos algorithm is given in Algorithm 4. The algorithm receives as input an optimal solution to the simplex relaxation, $\{u\}_{u\in V} \subseteq \Delta_k$, and outputs an assignment $f : V \to L$, assigning each vertex to a label in L.

[4] In order to obtain a relaxation for Uniform-Metric-Labeling, one needs to add to the objective a term corresponding to the assignment costs: $\sum_{u\in V}\sum_{i=1}^{k} c(u, i)\cdot u_i$.

Algorithm 4: Kleinberg-Tardos ($\{u\}_{u\in V} \subseteq \Delta_k$) [22]

```
while ∃u ∈ V s.t. u is unassigned:
    choose r ~ unif[0,1] and i ~ unif({1,2,...,k}).
    ∀u ∈ V s.t. u is unassigned and r ≤ u_i: f(u) ← i.
```

6.1 Uniform Metric Labeling via Algorithm 1

We observe that Algorithm 1 can be applied to the geometric relaxation of Uniform-Metric-Labeling, resulting in a new 2-approximation algorithm for the problem. The fact that the 2-approximation goes through for Uniform-Metric-Labeling follows from two observations. First, Lemma 1 can be applied, since the assumption that each edge differs in only two coordinates (Section 2) can still be made even though there are assignment costs. Second, by property (2b) of exponential clocks, for every $u \in \Delta_k$: $\Pr[\ell(u) = i] = u_i$. Hence, assignment costs are preserved in expectation. The above mentioned integrality gap instance of [22] for Uniform-Metric-Labeling (where the gap is $2(1 - 1/k)$) implies that our bound in Lemma 1 is indeed tight.

6.2 Multiway Cut via Algorithm 4

We observe that although the analysis of Algorithm 4 (as appearing in [22]) is asymptotically tight, it can still be tightened, proving that Algorithm 4 achieves the exact same guarantee as in Lemma 1. This implies that one can substitute the competing exponential clocks algorithm (Algorithm 1) with the algorithm of Kleinberg and Tardos for Uniform-Metric-Labeling (Algorithm 4) to obtain the improved approximation guarantees for Multiway-Cut. The slack in the proof of [22] results from upper bounding the separation probability of an edge (u, v) by the probability that u and v are assigned to labels in different iterations. Lemma 3 provides a tighter bound on the separation probability of an edge (u, v) by Algorithm 4, and its proof follows the same general outline as the proof of Lemma 1.

Lemma 3. *Let*

$$u = (u_1, u_2, u_3, \ldots, u_k) \text{ and}$$
$$v = (u_1 + \varepsilon, u_2 - \varepsilon, u_3, \ldots, u_k),$$

where $u, v \in \Delta_k$ for some $\varepsilon > 0$. Then,

$$\Pr[f(u) \neq f(v)] \leq \varepsilon(2 - u_1 - u_2).$$

Proof. Denote by B the event that both u and v are still unassigned at the end of an iteration, conditioning on u and v being unassigned at the start of the same iteration. For simplicity, we denote by r and i the random threshold and label chosen by Algorithm 4 in the same iteration. Let us calculate $\Pr[B]$:

$$\begin{aligned}\Pr[B] &= \sum_{j=1}^{k} \Pr[B|i = j] \cdot \Pr[i = j] \\ &= \frac{1}{k}\left(\Pr[r \geq u_1 + \varepsilon] + \Pr[r \geq u_2] + \ldots + \Pr[r \geq u_k]\right) \\ &= \frac{1}{k}\left((1 - (u_1 + \varepsilon)) + (1 - u_2) + \ldots + (1 - u_k)\right) \\ &= 1 - \frac{1+\varepsilon}{k}.\end{aligned}$$

For each $1 \leq i \leq k$, let A_i be the event that $f(u) = f(v) = i$. Let us calculate $\Pr[A_i]$, for all i, starting with $\Pr[A_1]$. Note that $\Pr[A_1]$ satisfies the following recursive formula:

$$\Pr[A_1] = \frac{u_1}{k} + \frac{\varepsilon \cdot u_1}{k} + \Pr[B] \cdot \Pr[A_1]. \quad (13)$$

The first term on the right hand side of (13) corresponds to the case that both u and v are assigned to label 1 in the current iteration, since:

$$\Pr[i = 1, r \leq \min\{u_1, u_1 + \varepsilon\}] = \frac{u_1}{k}.$$

The second term on the right hand side of (13) corresponds to the case that exactly one of u and v is assigned to label 1 in the current iteration and that the other is assigned to label 1 in some future iteration. Note that in this case it must be that v is assigned to label 1 in the current iteration and u is assigned to label 1 in some future iteration, since $u_1 < u_1 + \varepsilon = v_1$. We use the fact that the iterations are probabilistically independent and the marginal probability that any vertex u is assigned to label 1 at the end of the algorithm is u_1:

$$\Pr[i = 1, u_1 < r \leq u_1 + \varepsilon, f(u) = 1] =$$
$$\Pr[i = 1, u_1 < r \leq u_1 + \varepsilon] \cdot \Pr[f(u) = 1] = \frac{\varepsilon \cdot u_1}{k}.$$

The last and third term on the right hand side of (13) corresponds to the case that both u and v are not assigned to any label in the current iteration. As before, we use the fact that the iterations are probabilistically independent. Isolating $\Pr[A_1]$ in the recursive formula (13) yields that:

$$\Pr[A_1] = \frac{\frac{(1+\varepsilon)\cdot u_1}{k}}{1 - \Pr[B]} = \frac{\frac{(1+\varepsilon)\cdot u_1}{k}}{1 - \left(1 - \frac{1+\varepsilon}{k}\right)} = u_1.$$

The one before last equality is derived by plugging the value of $\Pr[B]$.

Let us calculate $\Pr[A_2]$. Note that $\Pr[A_2]$ satisfies the following recursive formula:

$$\Pr[A_2] = \frac{u_2 - \varepsilon}{k} + \frac{\varepsilon \cdot (u_2 - \varepsilon)}{k} + \Pr[B] \cdot \Pr[A_2]. \quad (14)$$

The first term on the right hand side of (14) corresponds to the case that both u and v are assigned to label 2 in the current iteration, since:

$$\Pr[i = 2, r \leq \min\{u_2 - \varepsilon, u_2\}] = \frac{u_2 - \varepsilon}{k}.$$

The second term on the right hand side of (14) corresponds to the case that exactly one of u and v is assigned to label 2 in the current iteration, and the other is assigned to label 2 in some future iteration. Note that in this case it must be that u is assigned to label 2 in the current iteration and v is assigned to label 2 in some future iteration, since $u_2 > u_2 - \varepsilon = v_2$. We use the fact that the iterations are probabilistically independent and the marginal probability that any vertex v is assigned to label 2 at the end of the algorithm is v_2:

$$\Pr[i = 2, u_2 - \varepsilon < r \leq u_2, f(v) = 2] =$$
$$\Pr[i = 2, u_2 - \varepsilon < r \leq u_2] \cdot \Pr[f(v) = 2] = \frac{\varepsilon \cdot (u_2 - \varepsilon)}{k}.$$

The last and third term on the right hand side of (14) corresponds to the case that both u and v are not assigned to any label in the current iteration. As before, we use the fact that the iterations are probabilistically independent. Isolating $\Pr[A_2]$ in the recursive formula (14) yields that:

$$\Pr[A_2] = \frac{\frac{(1+\varepsilon)\cdot(u_2-\varepsilon)}{k}}{1-\Pr[B]} = \frac{\frac{(1+\varepsilon)\cdot(u_2-\varepsilon)}{k}}{1-\left(1-\frac{1+\varepsilon}{k}\right)} = u_2-\varepsilon.$$

The one before last equality is derived by plugging the value of $\Pr[B]$.

Finally, let us calculate $\Pr[A_j]$ for all $j = 3, \ldots, k$. Note that $\Pr[A_j]$ satisfies the following recursive formula:

$$\Pr[A_j] = \frac{u_j}{k} + \Pr[B] \cdot \Pr[A_j]. \quad (15)$$

The first term on the right hand side of (15) corresponds to the case that both u and v are assigned to label j in the current iteration, since:

$$\Pr[i=j, r \le u_j] = \frac{u_j}{k}.$$

The second term on the right hand side of (15) corresponds to the case that both u and v are not assigned to any label in the current iteration. As before, we use the fact that the iterations are probabilistically independent. Note that the case that exactly one of u and v is assigned to label j in the current iteration cannot happen. Isolating $\Pr[A_j]$ in the recursive formula (15) yields that:

$$\Pr[A_j] = \frac{\frac{u_j}{k}}{1-\Pr[B]} = \frac{\frac{u_j}{k}}{1-\left(1-\frac{1+\varepsilon}{k}\right)} = \frac{u_j}{1+\varepsilon}.$$

The one before last equality is derived by plugging the value of $\Pr[B]$. The rest of the proof is identical to that of Lemma 1, thus, concluding the proof. □

6.3 Comparing Algorithms 1 and 4

Considering the fact that Algorithm 1 can be applied to obtain the 2-approximation for Uniform-Metric-Labeling, and Lemma 3 implies that Algorithm 4 can be applied to obtain the improved approximation guarantees for Multiway-Cut, one might ask whether both algorithms are equivalent. Formally, as both Algorithms 1 and 4 are randomized, the relevant question is whether the distributions over partitions of the simplex generated by Algorithms 1 and 4 are identical.

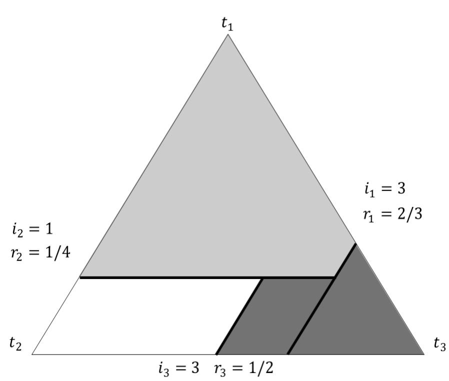

Figure 2: Label 3 (terminal t_3) is assigned a non-convex part of Δ_3. Denote by r_j and i_j the values of r and i in the jth iteration of Algorithm 4 [22].

The answer to the above question is negative for several reasons, implying that the algorithms are different. First, for example, there is a positive probability that Algorithm 4 assigns a non-convex part of the simplex to a label, as can be seen in Figure 2. However, the part of the simplex assigned to every label in Algorithm 1 is always convex. Second, Algorithm 4 uses cuts that are parallel to the faces of the simplex (as in [5, 18]), whereas Algorithm 1 utilizes cuts that are not parallel to the faces of the simplex (as can be seen in Figure 1).

7. OPEN PROBLEMS

An interesting open question is to extend our new simplex partitioning method to closely related problems. One possibility is the 0-Extension problem, for which there is a gap between the current best upper bound of $O(\log k/\log\log k)$ and the lower bound of $\Omega(\sqrt{\log k})$ (assuming the unique games conjecture).

Additionally, it still remains to close the gap between the (new) best algorithmic guarantee of 1.32388 presented in this paper and the best known integrality gap example of $8/(7+1/(k-1))$ for Multiway-Cut. This would settle the approximability of the problem assuming the unique games conjecture.

8. REFERENCES

[1] S. Arora, D. Karger, and M. Karpinski. Polynomial time approximation schemes for dense instances of np-hard problems. STOC '95, pages 284–293, 1995.

[2] S. Arora, S. Rao, and U. Vazirani. Expander flows, geometric embeddings and graph partitioning. *J. ACM*, 56(2):5:1–5:37, 2009.

[3] Y. Aumann and Y. Rabani. An $O(\log k)$ approximate min-cut max-flow theorem and approximation algorithm. *SIAM J. Comput.*, 27(1):291–301, 1998.

[4] G. Călinescu, H. Karloff, and Y. Rabani. Approximation algorithms for the 0-extension problem. SODA '01, pages 8–16, 2001.

[5] G. Călinescu, H. J. Karloff, and Y. Rabani. An improved approximation algorithm for multiway cut. *J. Comput. Syst. Sci.*, 60(3):564–574, 2000.

[6] C. Chekuri, S. Khanna, J. Naor, and L. Zosin. A linear programming formulation and approximation algorithms for the metric labeling problem. *SIAM J. Discrete Math.*, 18(3):608–625, 2004.

[7] J. Chuzhoy and J. Naor. The hardness of metric labeling. *SIAM J. Comput.*, 36(5):1376–1386, 2007.

[8] W. H. Cunningham and L. Tang. Optimal 3-terminal cuts and linear programming. IPCO '09, pages 114–125, 1999.

[9] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *SIAM Journal on Computing*, 23:864–894, 1994.

[10] G. Even, J. S. Naor, S. Rao, and B. Schieber. Divide-and-conquer approximation algorithms via spreading metrics. *J. ACM*, 47(4):585–616, 2000.

[11] J. Fakcharoenphol, C. Harrelson, S. Rao, and K. Talwar. An improved approximation algorithm for the 0-extension problem. SODA '03, pages 257–265, 2003.

[12] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *J. Comput. Syst. Sci.*, 69(3):485–497, 2004.

[13] A. Freund and H. J. Karloff. A lower bound of 8/(7+(1/k-1)) on the integrality ratio of the Calinescu-Karloff-Rabani relaxation for multiway cut. *Inf. Process. Lett.*, 75(1-2):43–50, 2000.

[14] A. M. Frieze and R. Kannan. The regularity lemma and approximation schemes for dense problems. FOCS '96, pages 12–20, 1996.

[15] N. Garg, V. V. Vazirani, and M. Yannakakis. Approximate max-flow min-(multi)cut theorems and their applications. *SIAM Journal on Computing*, 25:698–707, 1993.

[16] N. Garg, V. V. Vazirani, and M. Yannakakis. Multiway cuts in node weighted graphs. *J. Algorithms*, 50(1):49–61, 2004.

[17] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, Nov. 1995.

[18] D. R. Karger, P. N. Klein, C. Stein, M. Thorup, and N. E. Young. Rounding algorithms for a geometric embedding of minimum multiway cut. *Math. Oper. Res.*, 29(3):436–461, 2004.

[19] H. J. Karloff, S. Khot, A. Mehta, and Y. Rabani. On earthmover distance, metric labeling, and 0-extension. *SIAM J. Comput.*, 39(2):371–387, 2009.

[20] A. V. Karzanov. Minimum 0-extensions of graph metrics. *Eur. J. Comb.*, 19(1):71–101, 1998.

[21] S. Khot and O. Regev. Vertex cover might be hard to approximate to within 2-ε. *J. Comput. Syst. Sci.*, 74(3):335–349, May 2008.

[22] J. Kleinberg and E. Tardos. Approximation algorithms for classification problems with pairwise relationships: metric labeling and markov random fields. *J. ACM*, 49(5):616–639, 2002.

[23] T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *J. ACM*, 46(6):787–832, 1999.

[24] N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995.

[25] R. Manokaran, J. S. Naor, P. Raghavendra, and R. Schwartz. SDP gaps and UGC hardness for multiway cut, 0-extension, and metric labeling. STOC '08, pages 11–20, 2008.

[26] J. Naor and L. Zosin. A 2-approximation algorithm for the directed multiway cut problem. *SIAM J. Comput.*, 31(2):477–482, 2001.

[27] R. Y. Rubinstein and B. Melamed. *Modern Simulation and Modeling*. Wiley Series in Probability and Statistics, 1998.

APPENDIX

A. SET COVER AND COMPETING EXPONENTIAL CLOCKS

In this section we present a randomized rounding algorithm for the standard Set-Cover relaxation that succeeds with probability of 1 and is based on exponential clocks. The input for a Set-Cover instance consists of n elements $E = \{e_1, e_2, \ldots, e_n\}$, and m subsets $\mathcal{S} = \{S_1, \ldots, S_m\} \subseteq 2^E$. Each $S \in \mathcal{S}$ is associated with a cost c_S. The standard linear programming relaxation has a variable x_S for each set $S \in \mathcal{S}$, and the goal is to minimize $\sum_{S\in\mathcal{S}} c_S x_S$, while requiring that $\sum_{S:e\in S} x_S \geq 1$, $\forall e \in E$. Consider the following simple randomized rounding algorithm:

Algorithm 5: Set-Cover($\{x_S\}_{S\in\mathcal{S}}$)

1 choose i.i.d random variables $Z_S \sim \exp(1)$, $\forall S \in \mathcal{S}$.

2 output $\cup_{e\in E}$ argmin $\left\{\frac{Z_S}{x_S} \ : \ e \in S\right\}$.

The following lemma provides a bound on the probability that a set is chosen by Algorithm 5. Specifically, the probability that any set $S \in \mathcal{S}$ is chosen is at most $(\ln(|S|)+1)\cdot x_S$, providing an overall approximation of $(\ln(S_{\max}) + 1)$, where $S_{\max} = \max\{|S| \ : \ S \in \mathcal{S}\}$.

Lemma 4. *For every $S \in \mathcal{S}$, Algorithm 5 outputs S with probability of at most $(\ln(|S|) + 1) \cdot x_S$.*

Proof. Denote by $A_{S,e}$ the indicator for the event that $S = \text{argmin}\{Z_S/x_S \ : \ e \in S\}$, namely that element e chooses set S. Let $\alpha > 0$ be chosen later. Then,

$$\begin{aligned}\Pr[\exists e : A_{S,e}] &= \Pr[\exists e : A_{S,e} | Z_S/x_S < \alpha] \cdot \Pr[Z_S/x_S < \alpha] + \\ &\quad \Pr[\exists e : A_{S,e} | Z_S/x_S \geq \alpha] \cdot \Pr[Z_S/x_S \geq \alpha] \\ &\leq \Pr[Z_S/x_S < \alpha] + \\ &\quad \Pr[\exists e : A_{S,e} | Z_S/x_S \geq \alpha] \cdot \Pr[Z_S/x_S \geq \alpha] \\ &= \left(1 - e^{-\alpha\cdot x_S}\right) + \\ &\quad \Pr[\exists e : A_{S,e} | Z_S/x_S \geq \alpha] \cdot e^{-\alpha\cdot x_S}. \qquad (16)\end{aligned}$$

Applying the union bound over all elements in S and using the fact that $1 - e^{-x} \leq x$ for all x, one can upper bound (16) as follows:

$$\begin{aligned}&\left(1 - e^{-\alpha\cdot x_S}\right) + \Pr[\exists e : A_{S,e} \mid Z_S/x_S \geq \alpha] \cdot e^{-\alpha\cdot x_S} \leq \\ &\alpha \cdot x_S + e^{-\alpha\cdot x_S} \sum_{e\in S} \Pr[A_{S,e} \mid Z_S/x_S \geq \alpha]. \qquad (17)\end{aligned}$$

If $X \sim \exp(\lambda)$ and $Y \sim \exp(\mu)$ are independent, then $\forall t \geq 0$: $\Pr[X \leq Y \mid X \geq t] = \frac{\lambda}{\lambda+\mu} e^{-\mu t}$. The event $A_{S,e}$ is equivalent to $\frac{Z_S}{x_S} \leq \min\left\{\frac{Z_{S'}}{x_{S'}} \ : \ e \in S', S \neq S'\right\}$. By properties (1) and (2b) of the exponential distribution (Section 2) and the above, one can evaluate (17) as follows:

$$\begin{aligned}&\alpha \cdot x_S + e^{-\alpha\cdot x_S} \sum_{e\in S} \Pr[A_{S,e} \mid Z_S/x_S \geq \alpha] = \\ &\alpha \cdot x_S + e^{-\alpha\cdot x_S} \sum_{e\in S} \frac{x_S}{\sum_{S':e\in S'} x_{S'}} e^{-\alpha\left(\sum_{S':e\in S'} x_{S'} - x_S\right)} \leq \\ &\alpha \cdot x_S + |S| \cdot x_S \cdot e^{-\alpha} = \left(\alpha + |S| \cdot e^{-\alpha}\right) \cdot x_S.\end{aligned}$$

The inequality above is derived from the constraint that $\sum_{S:e\in S} x_S \geq 1$. By choosing $\alpha = \ln(|S|)$, one can conclude that: $\Pr[\exists e : A_{S,e}] \leq (\ln(|S|) + 1) \cdot x_S$. □