Approximation Schemes for Metric Bisection and Partitioning

W. Fernandez de la Vega* Marek Karpinski† Claire Kenyon‡

Abstract

We design *polynomial time approximation schemes* (PTASs) for Metric BISECTION, i.e. dividing a given finite metric space into two halves so as to minimize or maximize the sum of distances across the cut. The method extends to partitioning problems with arbitrary size constraints. Our approximation schemes depend on a hybrid placement method and on a new application of linearized quadratic programs.

1 Introduction.

MIN-BISECTION consists in dividing a graph into two equal halves so as to minimize the number of edges across the partition, and belongs to the most intriguing problems in the area of combinatorial optimization [16]. The reason is that we do not know at the moment how to deal with minimization global constraints such as partitioning the sets of vertices into two halves. Although there is currently no approximation hardness result for MIN-BISECTION (cf. [4, 19], see however [6]), the best known approximation factor is $O(\log^2 n)$ [7].

Here we consider the metric version of that problem: given a finite set V of points together with a metric, we ask for a partition of V into two equal parts such that the sum of the distances from the points of one part to the points of the other part is minimized. It is easy to see that metric MIN-BISECTION is NP-hard even if restricted to distances 1 and 2 (cf. [10]). In this paper we give a polynomial time approximation scheme (PTAS) for metric MIN-BISECTION and its k-ary size constraint generalizations. (This answers the open problems of [10].)

We draw on two lines of research to develop our algorithm. One is the method of "exhaustive sampling" for additive approximation for various optimization problems such as MAX-CUT or MAX-kSAT [2, 8, 14, 12, 13, 1]. The other connects to previous papers on approximation algorithms for metric problems and weighted dense problems [10, 9].

The rest of the paper is organized as follows. In Section 2, we formulate some metric and sampling lemmas. In Section 3, we construct our first PTAS for the metric MIN-BISECTION problem, which is purely combinatorial and extends [14]. In section 4, we use a non-smooth extension of a linear programming relaxation of [2]. Note that it is straightforward to adapt our algorithms to metric MAX-BISECTION, a problem which is relevant to clustering with cardinality constraints. In section 5, we give an extension to partitioning into two parts with size constraints $(k, n-k)$ (instead of $(n/2, n/2)$ for bisection), and a further extension to partitioning into a fixed number K of parts of prespecified sizes $(n_1, n_2, \ldots, n_K)$.

In the rest of the paper, we use the following notations. (V, d) denotes a finite metric space. For a subset U of V, and a vertex $v \in V$, we write $d(v, U) = \sum_{u \in U} d(u, v)$. For $A, B \subset V$, $d(A, B) = \sum_{u \in A, v \in B} d(u, v)$. Let $w_u = d(u, V)$, $W_U = \sum_{u \in U} w_u$, and $W = W_V$.

Our metric algorithms are partially inspired by existing algorithms for dense graphs, and can also be adapted to the dense graphs setting. What are the differences between the metric case and the dense graph case? Mainly, individual edges can have very large weights, inducing high variance problems. More precisely:

- In the metric setting, some vertices can have overwhelming importance (the ones which are very far from the rest and have weight close to W), and so we need to set those vertices aside and treat them separately. This cannot happen in dense graphs.
- In the metric setting, instead of doing a straightforward uniform sample, we need to perform a biased sample, where we give higher probability to vertices

*LRI, CNRS, Université de Paris-Sud, 91405 Orsay. Research partially supported by the IST grant 14036 (RAND-APX), and by IST APPOL2 and by the PROCOPE project. Email: lalo@lri.lri.fr

†Dept. of Computer Science, University of Bonn, 53117 Bonn. Research partially supported by DFG grants, DIMACS, PROCOPE grant 31022, IST grant 14036 (RAND-APX), and Max-Planck Research Prize. Research partially done while visiting Dept. of Computer Science, Yale University. Email: marek@cs.bonn.edu

‡LIX, Ecole Polytechnique. Research partically supported by the IST APPOL2 project. Email: kenyon@lix.polytechnique.fr

with high weight; this is necessary in order to get reliable estimates.

- In the metric setting, the estimate can be (with low probability) unacceptably large, thus we need to cap it to w_v. This does not happen in dense graphs.
- In the metric setting, the partition (V_j) must be done at random, whereas in dense graphs, one can take an arbitrary partition.
- In the metric setting the analysis no longer deals with sums of $\{0, 1\}$ variables (which describe the presence or absence of an edge in a graph); instead the terms in the sums can be quite large (since they describe distances); this makes the analysis of the variance much more delicate.
- Finally, in the metric setting our lower bound on OPT means that an additive error of $O(\epsilon W)$ implies a PTAS for the problem; that is not true for dense graphs.

We conjecture that metric MAX-BISECTION can be used in conjunction with [18] to design a PTAS for metric 2-clustering, subject to additional constraints on the cardinalities of the two clusters.
Most of the proofs are ommitted due to space constraints and will be included in the journal version of the paper.

2 Preliminary Results.

2.1 First attempt. One natural approach is to use random (suitably biased) sampling to estimate, for each point v, the sum of distances from v to each side of the optimal bisection, $d(v, L)$ and $d(v, R)$. For points which have about the same sum of distances to either side of the partition, it would intuitively seem that it does not matter on which side they are placed.
Unfortunately, this intuition is misleading, as the example in Figure 3 shows: we have four sets of vertices, A, B, C, D, each containing n vertices. All distances inside A, inside D, between A and B, and between C and D are equal to 1. All other distances are equal to 2.
It is not hard to check that on that input, the minimum bisection consists of the partition ($L = A \cup C, R = B \cup D$) and has value $\text{OPT} = 6n^2$.
For $v \in B$, $d(v, L) = 3n$ while $d(v, R) = 4n - 2$. Similarly for $v \in C$. Thus an estimator will easily be able to classify correctly the vertices of B and of C.
Notice that for $v \in A$, $d(v, L) = 3n - 1 \simeq 3n = d(v, R)$. Similarly, for $v \in D$, $d(v, R) = 3n - 1 \simeq 3n = d(v, L)$. Hence our sampling and estimating approach will consider all of these vertices to be equivalent and therefore place half of them on the left side and half of them on the right side, at random. This creates the bisection on the right hand side of Figure 3. The value of that bisection is: $13n^2/2$, which is a constant factor more than OPT.
This shows that, even if a vertex u is such that $d(u, L) \simeq d(u, R)$, it still matters where u goes.

2.2 Metric problems and lemmas. We now state lower bounds on the value of the optimal solution in the metric setting. (These are necessary for the problems in which the objective function must be minimized). First, an elementary metric lemma.

Proposition 2.1. ([11]) *Let $X, Y, Z \subseteq V$. Then* $|Z|d(X, Y) \leq |X|d(Y, Z) + |Y|d(Z, X)$.

In the $(k, n-k)$ Metric MIN-PARTITIONING problem, we are given a metric space (V, d) on n points and an integer $k < n$. The goal is to partition V into two sets of sizes k and $n-k$ so as to minimize the sum of distances across the partition. (Thus, MIN-BISECTION is the particular case of $k = n/2$.)
The lower bound below implies that, in order to get a PTAS for metric MIN-BISECTION, it suffices to obtain an additive approximation to within ϵW.
Let K be a fixed integer. Define the K-ary metric MIN-PARTITIONING problem as follows. Given a sequence of sizes $(n_1, n_2, \dots, n_K)$ such that $\sum_i n_i = n$, and given a finite metric space (V, d), find a partition of V into K parts of sizes $(n_1, n_2, \dots, n_K)$ so as to minimize the sum of distances between parts,

$$\sum_{u,v \text{ in different parts}} d(u, v).$$

Lemma 2.1. *(i) The optimal value of $(k, n-k)$ Metric* MIN-PARTITIONING *satisfies*

$$\text{OPT} \geq \frac{W}{2(1 + \frac{k}{n-k} + \frac{n-k}{k})}.$$

(ii) The optimal value of Metric MIN-BISECTION satisfies $\text{OPT} \geq W/6$.
(iii) Let ℓ be such that $(n_1 + \cdots + n_\ell) \leq n/2$. The optimal value of K-ary Metric MIN-PARTITIONING *for sizes $(n_1, n_2, \dots, n_K)$ satisfies*

$$\text{OPT} \geq \frac{W}{4} \frac{(n_1 + \cdots + n_\ell)}{n}.$$

Finally, the following metric Lemma will be useful in our analyses.

Lemma 2.2. ([10]) *$d(v, u) \leq 4w_v w_u / W$ for every u, v.*

2.3 Probabilistic lemmas. We recall, in the Lemma below, an inequality of Hoeffding (see also [15], Theorem 2.5, page 202).

LEMMA 2.3. ([17]) *Let (Y_i) be a sequence of independent random variables such that $0 \le Y_i \le b_i$ for every i. Let $Z = \sum_{1\le i\le n} Y_i$. Then, for any $a > 0$, we have*

$$\Pr(|Z - EZ| \ge a) \le 2e^{-2a^2/(\sum b_i^2)}.$$

LEMMA 2.4. *Let (Y_i) be a sequence of independent random variables and $Z = \sum_{1\le i\le n} Y_i$. Then:*

$$E(|Z - EZ|) \le \sqrt{\sum_i \sigma^2(Y_i)}.$$

Proof. $E(|Z - EZ|)^2 \le E((Z - EZ)^2) = \sigma^2(Z) = \sum_i \sigma^2(Y_i)$.

For $U \subset V$, the following lemma shows how to estimate $d(v, U)$ from a small biased sample of U.

LEMMA 2.5. (METRIC SAMPLING) *Let t be given and $U \subset V$. Let T be a random sample $\{u_1, u_2, ...u_t\}$ of U with replacement, where each u_i is obtained by picking a point $u \in U$ with probability w_u/W_U. Consider a fixed vertex $v \in V$. Then, with probability at least $1-2e^{-t\epsilon^2/8}$, we have:*

$$\left| d(v, U) - \frac{W_U}{t} \sum_{u\in T} \frac{d(v, u)}{w_u} \right| \le \epsilon d(v, U),$$

and moreover,

$$E(|d(v, U) - \frac{W_U}{t} \sum_{u\in T} \frac{d(v, u)}{w_u}|) \le \frac{2}{\sqrt{t}} d(v, U).$$

Proof. Consider the random variable $Z = \sum_{u\in T} d(v, u)/w_u$. We have:

$$Z = \sum_{i=1}^{t} Y_i,$$

where the Y_is are i.i.d.r.v.'s with

$$\forall u \in U, \quad \Pr\left(Y_i = \frac{d(v, u)}{w_u}\right) = \frac{w_u}{W_U}.$$

Y_i has average value $d(v, U)/W_U$ and maximum possible value at most $b_i = 4d(v, U)/W_U$ (by Lemma 2.2 applied to $U \cup \{v\}$). Applying Lemma 2.3 and scaling by W_U/t gives the first part of the lemma. The second part follows from Lemma 2.4, observing that any variable Y_i with range $[0, b_i]$ must have variance at most $b_i^2/4$.

LEMMA 2.6. *Let $s = 3/\epsilon^2$ be given and $U \subset V$. Let T be a random sample $\{u_1, u_2, ...u_s\}$ of U with replacement, where each u_i is obtained by picking a point $u \in U$ with probability w_u/W_U. and consider a partition of $U = (U_L, U_R)$. Assume that $W_{U_L} \ge W_{U_R}$. Then, with probability at least $1 - \epsilon$, we have $|S \cap U_L| \ge 1/\epsilon^2$.*

We will use the Metric Sampling Lemma jointly with exhaustive sampling. In our algorithms, the target U_L will be unknown; we will take a random biased sample S of a set which is larger than U_L, and try every possible subset T of S, so that, when we happen to try $T = S \cap U_L$, our subset T will be a biased sample of U_L.

3 A Combinatorial PTAS.

In this section we design and analyze a combinatorial PTAS for metric MIN-BISECTION. The method builds on the known metric sampling of [10] and hybrid placement techniques of [14].
The algorithm can be found in Figure 1. It takes as input a finite metric space (V, d). It makes a series of guesses and returns, when all these guesses are correct, a bisection of V whose cost is, with probability at least 3/4, at most $(1 + O(\epsilon))$OPT. The algorithm assumes that n is larger than some constant value, since for n small enough, one can just solve the problem by exhaustive search on V.

THEOREM 3.1. *With probability at least 3/4, the algorithm of Figure 1 computes a $(1 + O(\epsilon))$ approximation to Metric MIN-BISECTION. Its running time is $n^2 \cdot 2^{O(1/\epsilon^2)}$.*

3.1 A Preliminary Property.

We start with the following Lemma.

LEMMA 3.1. *Consider the partition constructed by the algorithm, $(B, V_1, \ldots, V_\ell)$. Consider the minimum partition of V, subject to the further constraint that it must be a bisection of* every *V_j. Then its expected value is at most $OPT + W\sqrt{\ell/n}$.*

Proof. The optimal bisection (L^*, R^*) induces a partition (L_j^*, R_j^*) of V_j. For each j, if $|L_j^*| > |R_j^*|$, we move $(|L_j^*| - |R_j^*|)/2$ random vertices from L_j^* to R_j^* (or vice-versa if $|L_j^*| < |R_j^*|$). This defines a bisection (L, R) satisfying the conditions of the lemma.
The expected number of points moved is $\sqrt{\ell n}$. The expected weight of the points moved is at most $W\sqrt{\ell/n}$.

3.2 Proof of Theorem 3.1.

The first part of the analysis is purely deterministic and, except for the last inequality, quite similar to the analysis in [14].

1. **Large weight vertices.** Let B denote the set of vertices with weight $> \epsilon^2 W/10$ and let $U = V \backslash B$.
2. **Sampling.** Let $s = 3/\epsilon^2$. Take a random sample S of U of size s obtained by independently drawing s points $u_1, u_2, ...u_s$ according to: $\Pr(u_1 = u) = w_u/W_U$ for $u \in U$.
3. **Exhaustive search.** Let $P_0 = (L, R)$ be an (unknown) near-optimal bisection. By exhaustive search[a], guess $B_L = B \cap L$ and $B_R = B \cap R$. Let $U_L = U \cap L$ and $U_R = U \cap R$ (U_L and U_R are not known). Assume that $W_{U_L} \geq W_{U_R}$. By exhaustive search, guess $T = S \cap U_L$. Let $t = |T|$. Moreover, by exhaustive search, guess $\widehat{W_{U_L}}$, the power of $(1+\epsilon)$ which is closest to W_{U_L}.
4. **Estimation.**

$$\forall v \in V, \text{ let } e_v = \min\{\frac{\widehat{W_{U_L}}}{t} \sum_{u \in T} \frac{d(v,u)}{w_u} + d(v, B_L), w_v\}. \tag{3.1}$$

5. **Partition.** Let $\ell = 1/\epsilon$ and define a partition $V_1, V_2, ..., V_\ell$ of U by placing each vertex in a V_j chosen uniformly at random (possibly moving one arbitrary vertex from each V_j to B if necessary so that the cardinality of V_j is even).
6. **Construction.** Let $A_0 = L_0$ and $B_0 = R_0$.

   For each $j = 1, 2, \ldots, \ell$, do the following:

   (a) **Estimation.** For each $v \in V_j$, let

   $$f_v = \sum_{k<j} d(v, A_k) + \frac{\ell - (j-1)}{\ell} e_v, \tag{3.2}$$

   $$\widehat{b}(v) = f_v - (w_v - f_v).$$

   f_v is an estimate on the distance from v to the left side of the hybrid partition, and $w_v - f_v$ is an estimate on the distance from v to the right side of the hybrid partition.

   (b) Construct a bisection (A_j, B_j) of V_j by placing the $|V_j|/2$ vertices with smallest value of $\widehat{b}(v)$ in B_j and placing the other $|V_j|/2$ vertices in A_j.

   Let $> \mathcal{A} = \cup_j A_j$ and $\mathcal{B} = \cup_j B_j$.
7. **Output.** Output the best of the bisections $(\mathcal{A}, \mathcal{B})$ thus constructed.

[a] meaning: we try every possibility, executing the rest of the algorithm for each of them. Each attempt results in a bisection. We will output the best one. The analysis will focus exclusively on the bisection constructed when the guess is correct.

Figure 1: A combinatorial algorithm for metric Minimum Bisection.

3.2.1 Part 1: Deterministic analysis. Let P_j be the following hybrid bisection:

$$(\bigcup_{k<j} A_j \cup \bigcup_{k\geq j} L_j, \bigcup_{k<j} B_j \cup \bigcup_{k\geq j} R_j) = (\text{Left}(P_j), \text{Right}(P_j)).$$

The output is P_ℓ, and $\text{COST}(P_\ell) - \text{COST}(P_0)$ can be rewritten as

$$\sum_{1\leq j\leq \ell} [\text{COST}(P_j) - \text{COST}(P_{j-1})].$$

Consider the vertices which are classified differently in P_{j-1} and in P_j: there is a subset $X = \{x_1, \ldots, x_m\}$ of L_j and a subset $Y = \{y_1, \ldots, y_m\}$ of R_j, of the same cardinality, such that $A_j = L_j - X + Y$ and $B_j = R_j - Y + X$. For each vertex u, let $b(u) = d(u, \text{Left}(P_{j-1})) - (w_u - d(u, \text{Left}(P_{j-1})))$. We have:

$$\begin{aligned} &\text{COST}(P_j) - \text{COST}(P_{j-1}) \\ &\quad\leq \sum_{x_i\in X} b(x_i) - \sum_{y_i\in Y} b(y_i) + 2\sum_{X\times Y} d(x,y) \\ &\quad\leq \sum_{1\leq i\leq m} (b(x_i) - b(y_i)) + 2d(V_j, V_j). \end{aligned}$$

Now, here is the central part of the proof:

$$\begin{aligned} &b(x_i) - b(y_i) \\ &\quad= (b(x_i) - \widehat{b}(x_i)) + (\widehat{b}(x_i) - \widehat{b}(y_i)) + (\widehat{b}(y_i) - b(y_i)) \\ &\quad\leq (b(x_i) - \widehat{b}(x_i)) + (\widehat{b}(y_i) - b(y_i)), \end{aligned}$$

since x_i is placed to the right and y_i is placed to the left, and so by definition of the algorithm it must be that $\widehat{b}(x_i) \leq \widehat{b}(y_i)$. Thus

$$\begin{aligned} &\text{COST}(P_j) - \text{COST}(P_{j-1}) \\ &\quad\leq \sum_{u\in V_j} |b(u) - \widehat{b}(u)| + 2d(V_j, V_j) \\ &\quad\leq 2\sum_{u\in V_j} |\sum_{k\geq j} d(u, L_k) - \frac{\ell-(j-1)}{\ell}(e_u - d(u, B_L))| \\ &\qquad\qquad + 2d(V_j, V_j). \end{aligned}$$

Now,

$$\begin{aligned} &|\sum_{k\geq j} d(u, L_k) - \frac{\ell-(j-1)}{\ell}(e_u - d(u, B_L))| \\ &\quad\leq |\sum_{k\geq j} d(u, L_k) - \frac{\ell-(j-1)}{\ell} d(u, U_L)| + \\ &\qquad\quad \frac{\ell-(j-1)}{\ell} |d(u, U_L) - (e_u - d(u, B_L))|. \end{aligned}$$

We must now use probabilistic tools to analyze this equation.

3.2.2 Part 2: Probabilistic analysis. Let us analyze the first term of the right hand side of the equation. Fix $v \in V_j$ and let $Z_v = \sum_{k\geq j} d(v, L_k)$. The expectation of Z_v is $d(v, U_L)(ell - j + 1)/ell$, and so we must analyze $|Z_v - EZ_v|$. We have: $Z_v = \sum_{u\in U_L} d(v,u)X_u$, where the X_u are i.i.d.r.v.'s, with X_u equal to 1 with probability $(\ell-(j-1))/\ell$ and to 0 with the complementary probability.

We split Z_v into two parts, $Z_v = A_v + B_v$, with

$$\begin{cases} A_v = \sum_{u: d(u,v)\leq w_v\epsilon/\sqrt{n}} d(u,v)X_u \\ B_v = \sum_{u: d(u,v)> w_v\epsilon/\sqrt{n}} d(u,v)X_u. \end{cases}$$

The first of these two parts is straightforward: applying Lemma 2.4 to A_v, with $b_i = w_v\epsilon/\sqrt{n}$, yields

$$E(|A_v - EA_v|) \leq \epsilon w_v/2.$$

For the second part, from Proposition 2.1 for $X = \{u\}, Y = \{v\}$, $Z = V$, we get $nd(u,v) \leq w_u + w_v$, so $d(u,v) > w_v\epsilon/\sqrt{n}$ implies that $w_u > (\epsilon\sqrt{n}-1)w_v$. Thus $d(u,v) \leq (w_u + w_v)/n \leq 2w_u/n$. Applying Lemma 2.4 to B_v, with $b_u = 2w_u/n$, now yields

$$E(|B_v - EB_v|) \leq \frac{\sqrt{\sum_u w_u^2}}{n}.$$

Since $\sum w_u \leq W$ and $\max w_u \leq \epsilon^2 W$, we have $\sum w_u^2 \leq \epsilon^2 W^2$, and so

$$E(|B_v - EB_v|) \leq \frac{\epsilon W}{n}.$$

Summing gives

$$E(|Z_v - EZ_v|) \leq \frac{\epsilon w_v}{2} + \frac{\epsilon W}{n}.$$

As for the second term of the equation, we first let

$$e'_v = \min\{\frac{W_{U_L}}{t}\sum_{u\in T}\frac{d(v,u)}{w_u} + d(v, B_L), w_v\}.$$

From Lemma 2.5 applied to U_L, we have:

$$E(|d(v, U_L) - (e'_u - d(u, B_L))|) \leq \frac{2}{\sqrt{t}} d(v, U_L) \leq \frac{2}{\sqrt{t}} w_v.$$

Since our estimate for W_{U_L} is within a $(1+\epsilon)$ factor of the actual value, we moreover have:

$$E(|e'_u - e_u|) \leq \epsilon w_u.$$

The rest of the proof is easy, plugging these bounds into Equation 3.3, summing over j, using Lemmas 2.6 and 3.1, Markov's inequality, and Lemma 2.1.

■

1. **Large weight vertices.** Let B denote the set of vertices v with $w_v \geq \epsilon^2 W/100$, and let $U = V \setminus B$.
2. **Sampling.** Let $s = 3/\epsilon^2$. Take a random sample S of U of size s obtained by independently drawing s points $u_1, u_2, ...u_s$ according to: $\Pr(u_1 = u) = w_u/W_U$ for $u \in U$.
3. **Exhaustive search.** Let (L, R) be the (unknown) optimal bisection. By exhaustive search, guess $B_L = B \cap L$ and $B_R = B \cap R$. Let $\Delta = \sum_{B_L \times B_R} d(u, v)$. Let $U_L = U \cap L$ and $U_R = U \cap R$ (U_L and U_R are not known). Assume that $W_{U_L} \geq W_{U_R}$. By exhaustive search, guess $T = S \cap U_L$. Let $t = |T|$. Moreover, by exhaustive search, guess $\widehat{W_{U_L}}$, the power of $(1+\epsilon)$ which is closest to W_{U_L}.
4. **Estimation.**

(3.3) $$\forall v \in V, \text{ let } e_v = \min\{\frac{\widehat{W_{U_L}}}{t} \sum_{u \in T} \frac{d(v,u)}{w_u} + d(v, B_L), w_v\}.$$

5. **Construction.**

(a) Let $c(x) = \sum_{v \in U} x_v e_v + \sum_{v \in U}(1 - x_v) d(v, B_R) + \Delta$. Solve the following linear program $LP(n)$ with variables x_v and z_v, $v \in U$,

Minimize $c(x)$ s.t.

$$\begin{cases} \forall v, \quad 0 \leq x_v & \leq \quad 1 \\ \forall v, \quad d(v, B_L) + \sum_{u \in U}(1 - x_u) d(u, v) & \leq \quad e_v + z_v \\ \forall v, \quad d(v, B_L) + \sum_{u \in U}(1 - x_u) d(u, v) & \geq \quad e_v - z_v \\ \quad \sum_v z_v & \leq \quad 30\epsilon W \\ \quad \sum_v x_v + |B_L| & = \quad n/2. \end{cases}$$

Let (x_v^*, z_v^*) denote the optimal fractional solution.

(b) Use randomized rounding to obtain an integer vector (y_v): for every v independently, y_v is set to 1 with probability x_v^* and to 0 with the complementary probability. Together with (B_L, B_R), this defines a partition of V.

(c) Repair the unbalance by moving from the side with the larger size to the other side the required number of vertices with smallest weights.

6. **Output.** Output the best of the bisections thus constructed.

Figure 2: A linear programming algorithm for metric Minimum Bisection.

Remarks.

1. It is not necessary to take the number of parts V_j exactly $\ell = 1/\epsilon$. The algorithm could be adapted to work for any number $\ell \in [1/\epsilon, n\epsilon^2]$.

2. Except for biased sampling, which is specific to the metric situation, the additional ideas used here to modify the hybrid placement technique from [14] could be applied to the dense graphs setting as well. This would improve the query complexity from [14] by a factor of $O(\ln(1/\epsilon)/\epsilon)$.

4 A PTAS Based on Linear Programming.

In this part we combine exhaustive search on the points with highest weights, biased sampling, and give a new non-smooth extension of the linearization approach of [2]. In addition, we modify the LP approach slightly (by introducing n new variables z_v) in such a way that one can compute estimates by taking samples of size $O_\epsilon(1)$ only (instead of $O(\log n)$). (We believe that this improvement could also be applied to the algorithms of [2].)

We represent a bipartition (S, T) of V by the vector (x_v) where $x_v = 0$ if $v \in S$, and $x_v = 1$ if $v \in T$. We denote by (L, R) an optimum bisection. For each vertex v, e_v will be an estimator for $d(v, L)$.

If n is smaller than some constant depending on ϵ (see proof of lemma 4.5), we solve by exhaustive search. Otherwise, we run the algorithm presented on Figure 2 at the end of the paper. Throught this section we will refer to the notation used in the description of this algorithm.

THEOREM 4.1. *With probability at least* $3/4$*, the algorithm in Figure 2 computes a* $(1 + O(\epsilon))$ *approximation to metric MIN-BISECTION. Its running time is* $LP(n)2^{O(1/\epsilon^2)}$*, where* $LP(n)$ *denotes the running time to solve a linear program with* $O(n)$ *underlying variables and constraints.*

LEMMA 4.1. *1. Let* $S = \sum_{v \in U} |d(v, L) - e_v|$*. Then, with probability at least* $89/100$*, we have* $ES \leq 30\epsilon W$*.*

2. Let (x_v^*, z_v^*) *denote the optimal fractional solution of the linear program, and* (y_v) *denote the result of the randomized rounding. Then* $E(\sum_{v \in U} |x_v^* - y_v| w_v) \leq \epsilon W/20$*.*

3. $E(|\sum_{v \in U}(x_v^* - y_v)|) \leq \sqrt{n}/2$*.*

Let (x_v) be the optimal bisection and (x_v^*, z_v^*) the optimal fractional solution of the linear program.

LEMMA 4.2. *With probability at least* $89/100$*, the optimal bisection* (x_v) *is feasible, and moreover*

$$OPT = \mathrm{COST}(x_v) \geq c(x_v^*) - 30\epsilon W.$$

Let (y_v) denote the partition obtained by the randomized rounding.

LEMMA 4.3. $E(c(y)) \leq E(c(x^*)) + \epsilon W/10$.

LEMMA 4.4.

$$E(\mathrm{COST}(y)) \leq E(c(y)) + \epsilon W/10 + E(\sum_v z_v^*).$$

Let (y'_v) denote the bisection output by the algorithm.

LEMMA 4.5.

$$E(\mathrm{COST}(y'_v)) \leq E(\mathrm{COST}(y_v)) + \frac{W}{2\sqrt{n}}.$$

To prove Theorem 4.1, we combine Lemmas 4.5, 4.4, 4.3 and 4.2.

The running time follows by inspection. ■

Which algorithm is better? To a large degree, which of the two algorithms is preferable is a matter of taste. The algorithms have many features in common, which we have tried to highlight. The first algorithm is purely combinatorial, is entirely self-contained, and is the winner in terms of running time as a function of n. The second algorithm is a reduction to linear programming, much easier to generalize to variants of the problem (simply by suitably modifying the linear program), and is thus more robust.

5 Extensions.

We note that both algorithms in sections 3 and 4 can be adapted to construct much more efficient algorithms for the problem of Metric MAX-CUT [10].

THEOREM 5.1. *There is a PTAS for Metric MAX-CUT with running time* $O(n^2 \cdot 2^{O(1/\epsilon^2)})$*.*

We recall from section 2.2 the following definition of the $(k, n-k)$ Metric MIN-PARTITIONING problem: we are given a metric space (V, d) on n points and an integer $k < n$. The goal is to partition V into two sets with sizes k and $n - k$ so as to minimize the sum of distances across that partition.

THEOREM 5.2. *The problem of* $(k, n-k)$ *Metric MIN-PARTITIONING has a PTAS.*

Proof. We present the algorithm.

1. If $k/n \geq \epsilon/2$, then we apply the algorithm on Figure 2 once for $\epsilon' = \epsilon^2$, replacing the last constraint of the linear program by $\sum_v x_v + |B_L| = k$, and modifying the last step to repair the unbalance is done by moving from appropriately so as to obtain a $(k, n-k)$ partition; and once for $\epsilon' = \epsilon^2$, replacing the last constraint of the linear program by $\sum_v x_v + |B_L| = n-k$, and modifying the last step to repair the unbalance is done by moving from appropriately so as to obtain an $(n-k, k)$ partition. We output the better of those two partitions.
2. If $k/n < \epsilon/2$,then let S denote the k vertices with smallest weight. We output the partition $(S, V \setminus S)$.

The analysis is ommitted.

Let K be a fixed integer. Define the K-ary metric MIN-PARTITIONING as follows. Given a sequence of sizes $(n_1, n_2, \ldots, n_K)$ such that $\sum_i n_i = n$, and given a finite metric space (V, d), find a partition of V into K parts of sizes $(n_1, n_2, \ldots, n_K)$ so as to minimize the sum of distances between parts,

$$\sum_{u,v \text{ in different parts}} d(u, v).$$

THEOREM 5.3. *There is a PTAS for K-ary metric MIN-PARTITIONING.*

Proof. See appendix for the algorithm (analysis ommitted).

We consider now another applications towards the problems of MIN-k-CUT, and MIN-MULTIWAY-CUT (cf. [23]), [5]) embedded in a metric space.

Metric MIN-k-CUT is the problem of partitioning a given finite (V, d) space metric into k parts as to minimize the sums of distances between different parts. Metric MIN-MULTIWAY-k-CUT is the problem, given a finite metric (V, d) and a set of k terminals $T \subseteq V$, to partition (V, d) as to disconnect every terminal from each other and to minimize the sums fo distances between different parts.

The methods from section 4 can be easily adopted to yield the following

THEOREM 5.4. *There are PTASs for Metric MIN-k-CUT and Metric MIN-MULTIWAY-k-CUT.*

Acknowledgements. We thank Mark Jerrum, Uri Feige, Alan Frieze, and Ravi Kannan for stimulating discussions. We thank also Yuval Peres for pointing out Lemma 2.4.

References

[1] N. Alon, W. Fernandez de la Vega, R. Kannan, and M. Karpinski, *Random Sampling and MAX-CSP Problems*, Proc. 34th ACM STOC (2002), pp. 232-239.

[2] S. Arora, D. Karger, and M. Karpinski, *Polynomial Time Approximation Schemes for Dense Instances of NP-Hard Problems*, Proc. 27th STOC (1995), pp. 284-293; J. Comput. System Sciences 58 (1999) 193-210.

[3] G. Bennet,*Probability inequalities for sums of independent random variables*, Journal of the American Statistical Association 57 (1962) 33-45

[4] P. Berman and M. Karpinski, *Approximation Hardness of Bounded Degree MIN-CSP and MIN-BISECTION*, ECCC Technical Report, TR01-026, 2001, also in Proc. 29th ICALP (2002), LNCS 2380, Springer, 2002, pp. 623-632.

[5] E. Dahlhaus, D.S. Johnson, C.H. Papadimitriou, P.D. Seymour and M. Yannakakis, *The Complexity of Multiterminal Cuts*, SIAM J. Comput 23 (1994), pp. 864-894.

[6] U. Feige, Relations between Average Case Complexity and Approximation Complexity, Proc. 34^{th} ACM STOC (2002), pp. 534-543.

[7] U. Feige and R. Krauthgamer, *A Polylogarithmic Approximation of the Minimum Bisection*, Proc. 41st IEEE FOCS (2000), pp. 105–115.

[8] W. Fernandez de la Vega, *MAX-CUT has a Randomized Approximation Scheme in Dense Graphs,* Random Structures and Algorithms 8 (1996), pp.187-198

[9] W. Fernandez de la Vega and Marek Karpinski, *Polynomial time approximation of dense weighted instances of MAX-CUT*, Random Structures and Algorithms 16 (2000), pp. 314-332.

[10] W. Fernandez de la Vega and Claire Kenyon, *A randomized approximation scheme for metric MAX-CUT*, Proc. 39^{th} IEEE FOCS (1998), pp. 468-471, final version in J. Comput. System Sciences 63 (2001), pp. 531-541.

[11] W. Fernandez de la Vega, M.Karpinski, C. Kenyon, and Y. Rabani, *Approximation schemes for clustering problems*, Proc. 35th ACM STOC(2003), to appear.

[12] A.M. Frieze and R. Kannan, *The Regularity Lemma and Approximation Schemes for Dense Problems,* Proc. 37th FOCS (1996), pp. 12-20.

[13] A.M. Frieze and R. Kannan, *Quick Approximation to Matrices and Applications,* Combinatorica 19 (2) (1999), pp. 175-120.

[14] O. Goldreich, S. Goldwasser and D. Ron, *Property Testing and its Connection to Learning and Approximation,* Proc. 37th IEEE FOCS (1996), pp. 339-348; J. ACM 45 (1998), pp. 653-750.

[15] M. Habib, C. McDiarmid, J. Ramirez-Alfonsin, B. Reed (eds.), *Probabilistic Methods for Algorithmic Discrete Mathematics*, Concentration, Springer, 1998.

[16] D.S. Hochbaum(ed.), Approximation Algorithms for NP-Hard Problems, PWS, 1997.

[17] W. Hoeffding, *Probability inequalities for sums of bounded random variables*, Journal of the American Statistical Association 58 (1963) 13-30.
[18] P. Indyk, *A Sublinear Time Approximation Scheme for Clustering in Metric Spaces,* Proc. 40th IEEE FOCS (1999), 154-159.
[19] M. Karpinski, Approximability of the Minimum Bisection Problem, Proc. 27^{th} MFCS (2002), LNCS Vol. 2420, Springer, 2002, pp. 59-67.
[20] C. McDiarmid, *Concentration*, Probabilistic Methods for Algorithmic Disctrete Mathematics, M.Habib, C.McDiarmid, J. Ramirez Alfonsin, B.Reed (Eds.), Springer 1998.
[21] C. H. Papadimitriou and M. Yannakakis. *Optimization, Approximation and Complexity Classes*, J. Comput. System Sciences, 43 (1991), pp. 425-440.
[22] L.J. Schulman. Clustering for edge-cost minimization. Proc. 32nd ACM STOC (2000), pp. 547–555.
[23] H. Saran and V. V. Vazirani *Finding k-Cuts within Twice the Optimal*, Proc. 32nd IEEE FOCS (1991), pp. 743-751.

A The algorithm for Size Constraint Metric MIN-PARTITIONING

If n is smaller than some constant depending on ϵ, we solve by exhaustive search. Otherwise, we use the following algorithme, which is an extension of our linear programming algorithm for $(k, n-k)$ MIN-PARTITIONING.

1. Let $n_1, n_2, \ldots, n_\ell$ denote the sizes smaller or equal to $\epsilon^2 n/K$ and $m = n_1+n_2+\cdots+n_\ell$. Take the the m vertices with smallest weights, and partition them arbitrarily to create ℓ parts of sizes $n_1, n_2, \ldots, n_\ell$.

2. By exhaustive search, guess the cardinalities of the parts with index $\geq \ell+1$ and with weight $\leq \epsilon^2 W/K$ in an optimum solution. Up to renaming, we can assume that these are the last h parts. Let $r = n_{K-h+1} + n_{K-h+2} ... + n_K$. Take the the r remaining vertices with smallest weights, and partition them arbitrarily to create h parts of sizes $n_{K-h+1}, n_{K-h+2}, \ldots, ..., n_K$. Let V' denote the remaining set of vertices.

3. In what follows, we extend the algorithm on figure 2 to the metric MIN-PARTITIONING problem with constraints
$$n_{\ell+1}, n_{\ell+2}, ..., n_{K-h}$$
and vertex set V'. We rename the constraints as $(n_1, n_2, ...n_{K'})$ with a new $K' = K = \ell - h$. We call this the *reduced* problem.

4. Let B denote the vertices with weight $\geq \epsilon^2 W/100$ and $U = V' \setminus B$.

5. Take a random biased sample S of U of size $s = O(1/\epsilon^4)$. (Note the change of the value of s compared to its value in the algorithm of figure 2.)

6. By exhaustive search, guess the partition $(B_1, B_2, \ldots B_K)$ of B induced by the optimal solution. Let $\Delta = \sum_{i\neq j} d(B_i, B_j)$. For each $i \in \{1, \ldots, K'\}$, guess the intersection T_i of S with the i^{th} part of the optimal partition, of size t_i. Moreover, by exhaustive search, guess the approximate weight $\widehat{W}_i$ of that part.

7. For each $v \in U$ and for each i, let
$$e_{v,i} = \min\{\frac{\widehat{W}_i}{t_i} \sum_{u\in T_i} \frac{d(u,v)}{w_u} + d(v, B_i), w_v\}.$$

8. Let $c(x) = \sum_{v\in U}\sum_i x_{v,i}(\sum_{k\neq i} e_{v,k}) + \sum_{v,i}(1 - x_{v,i})d(v, B_i) + \Delta$. Solve the following linear program with variables $x_{v,i}$ and $z_{v,i}$, $v \in U$ and $i \leq K'$:
$$\min c(x)$$
subject to the constraints
$$\begin{cases} \forall v, i \ \ x_{v,i} & \geq \ 0 \\ \forall v, \ \ \sum_i x_{v,i} & = \ 1 \\ \forall v, i \ \ d(v, B_i) + \sum_{u\in U} x_{u,i} d(u, v) & \leq \ e_{v,i} + z_{v,i} \\ \forall v, i \ \ d(v, B_i) + \sum_{u\in U} x_{u,i} d(u, v) & \geq \ e_{v,i} - z_{v,i} \\ \quad \sum_i \sum_v z_{v,i} & \leq \ 30\epsilon^2 W \\ \forall i, \ \ |B_i| + \sum_{v\in U} x_{v,i} & = \ n_i \end{cases}$$
Let $(x^*_{v,i}, z^*_{v,i})$ denote the optimal fractional solution.

9. Use randomized rounding to obtain an integer vector $(y_{v,i})$: for every v independently, choose an i according to the distribution defined by $(x^*_{v,i})_i$, and set that $y_{v,i}$ to 1 and the others to 0. Together with $(B_1, \ldots, B_{K'})$, this defines a partition $P = C_{\ell+1}, C_{\ell+2}, ...C_{K-h}$ of V'.

10. Ajust the sizes analogously to the last step of the linear programming MIN-BISECTION algorithm to get a partition P' with part sizes $|C'_{\ell+1}| = n_{\ell+1}, |C'_{\ell+2}| = n_{\ell+2}, ...|C'_{K-h} = n_{K-h}$.

11. Complete P' by the parts defined in items 1 and 2 to get the output partition P''.

This ends the description of the algorithm.

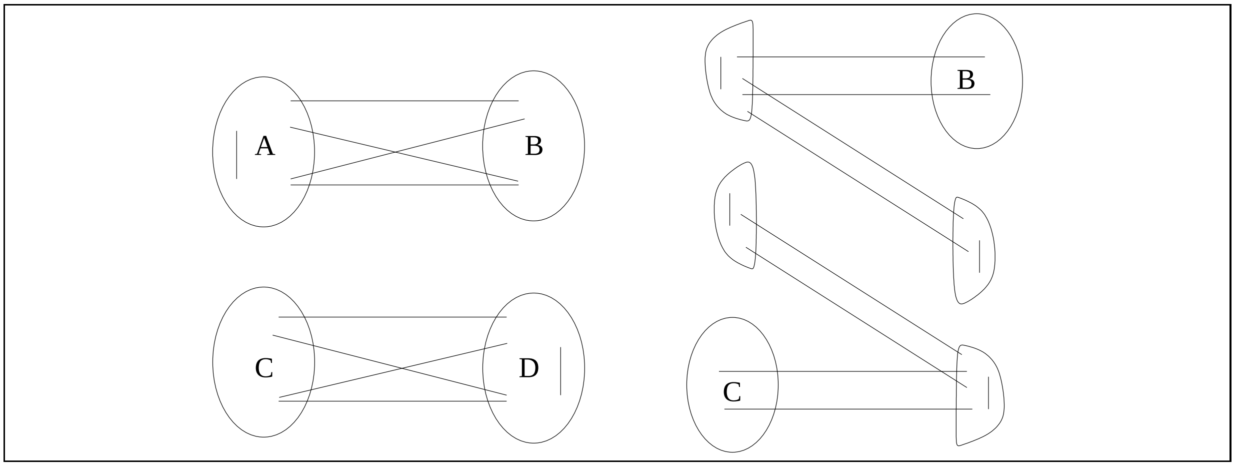

Figure 3: An example showing why, even if we have a reliable estimate of $d(v, L)$ and of $d(v, R)$ for every v, that is not sufficient to construct a near-optimal partition in the natural manner.

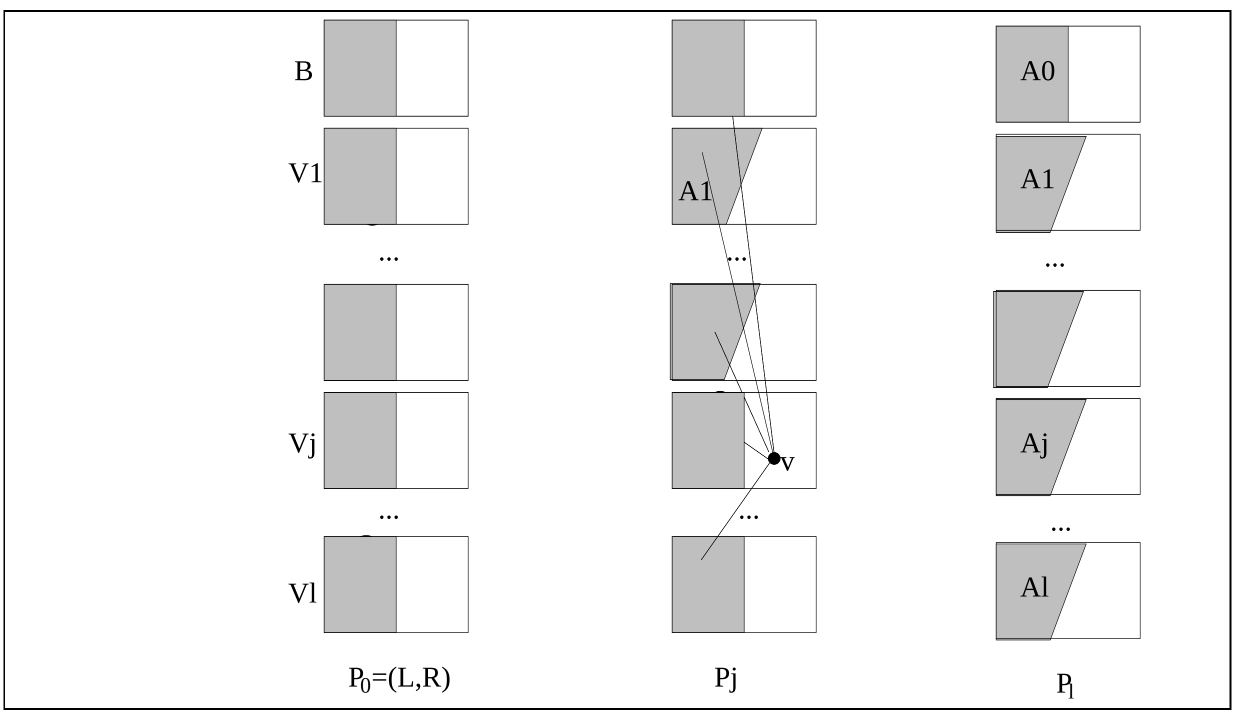

Figure 4: The hybrid partitions used by the combinatorial algorithm. f_v is an estimate of $d(v, \text{Left}(P_j))$ for $v \in V_j$.