Approximation Schemes for Metric Bisection and Partitioning

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

November 10, 2003

Abstract.
We design *polynomial time approximation schemes* (PTASs) for Metric MIN-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 [H97]. The reason is that we do not know at the moment how to deal with the minimization global conditions such as partitioning the sets of vertices into two halves. Although there is currently no approximation hardness result for MIN-BISECTION (cf. [BK01, K02], see however [F02]), the best known approximation factor is $O(\log^2 n)$ [FK00].

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. [FK98]). In this paper we give a polynomial time approximation scheme (PTAS) for metric MIN-BISECTION. (This answers an open problem from [FK98].)

We draw on two lines of research to develop our algorithm. One is a method of "exhaustive sampling" for additive approximation for various optimization problems such as MAX-CUT or MAX-kSAT [AKK95, F96, GGR96, FK96, FK97, AFKK02]. The other connects to the previous papers on approximate algorithms for metric problems and weighted dense problems [FK98, FVK00].

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 [GGR96]. In section 4, we use a non-smooth extension of a linear programming relaxation of [AKK95]. Note that it is straightforward to adapt our algorithms to the Maximum Bisection problem. In section 6, we give an extension to partitioning into two parts of prespecified sizes $(k, n-k)$ so as to minimize the distances accross the cut, and a further extension to "size constraint MIN PARTITIONING" problems, where the goal is to partition into a fixed number K of parts of prespecified sizes $(n_1, n_2, \ldots, n_K)$, so as to minimize the sum of distances between the points which are placed in different parts.

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 $W = W_V$.

*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`

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 1 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.

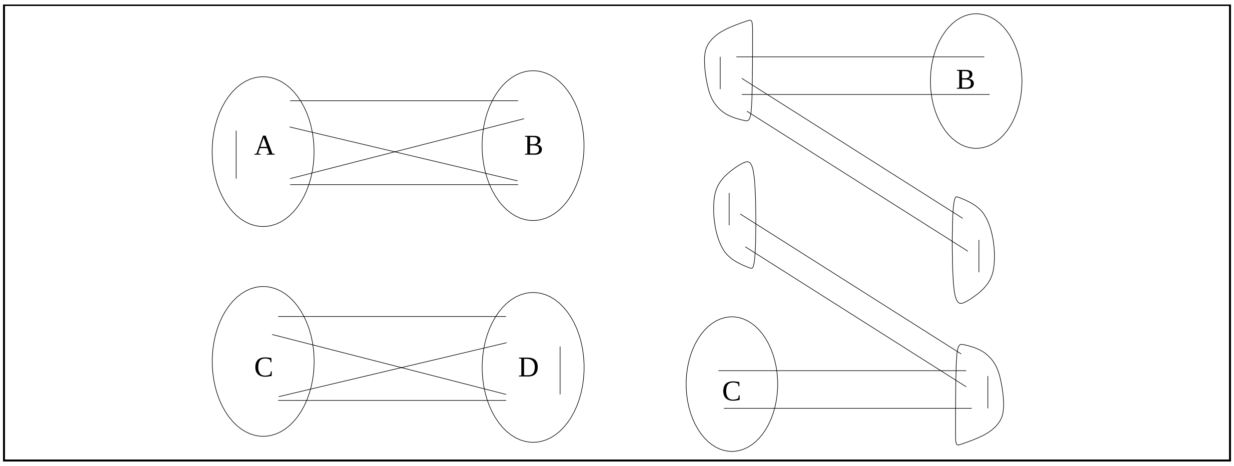

Figure 1: An example showing why, even if we have a reliable estimate of $d(v, L)$ and of $d(v, R)$ for every v, that is not sufficient to construct a near-optimal partition in the natural manner.

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 $\mathrm{OPT} = 6n^2$.
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 1. 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 lemmas

We are going to formulate some metric results used in this paper.

Proposition 1 ([FKKR03]) *Let* $X, Y, Z \subseteq V$. *Then* $|Z| d(X, Y) \le |X| d(Y, Z) + |Y| d(Z, X)$.

Lemma 1 ([FK98]) $d(v, u) \le 4 w_v w_u / W$ *for every* u, v.

The following lemma implies that, in order to get a PTAS for metric MIN-BISECTION, it suffices to obtain an additive approximation to within ϵW.

Lemma 2 *The optimal value of Metric MIN-BISECTION satisfies* $\mathrm{OPT} \ge W/6$.

Proof: Let (L, R) be the optimal min bisection. Apply Proposition 1 to $X = Y = L, Z = R$ and to $X = Y = R, Z = L$ to get

$$\begin{cases} 2(n/2)OPT & \geq & (n/2)d(L, L) \\ 2(n/2)OPT & \geq & (n/2)d(R, R). \end{cases}$$

Thus $W = d(L, L) + d(R, R) + 2d(L, R) \leq 6OPT$.

■

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 with sizes k and $n - k$ so as to minimize the sum of distances across that partition. (Thus, MIN-BISECTION is the particular case of $k = n/2$.) Lemma 2 generalizes to Metric MIN-PARTITIONING as follows.

Lemma 3 *The optimal value of $(k, n-k)$ Metric* MIN-PARTITIONING *satisfies*

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

Proof: Apply Proposition 1 again to $X = Y = L, Z = R$ and to $X = Y = R, Z = L$ to get

$$d(R, L) \geq \max\left(\frac{k}{n-k-1} d(R, R), \frac{n-k}{k-1} d(L, L)\right)$$

The maximum is minimized by equalizing the two terms on the right hand-side of the above, hence the Lemma.

■

2.3 Sampling lemmas

We recall, in the Lemma below, an inequality of Hoeffding (see also [HM98], Theorem 2.5, page 202).

Lemma 4 ([H63]) *Let (Y_i) be a sequence of independent random variables such that $0 \leq Y_i \leq b_i$ for every i. Let $Z = \sum_{1 \leq i \leq n} Y_i$. Then, for any $a > 0$, we have*

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

Corollary 1

$$E(|Z - EZ|) \leq \sqrt{\frac{\pi \sum b_i^2}{2}}.$$

Proof: $E(|Z - EZ|) = \int \Pr(|Z - EZ| > x) dx \leq \int 2e^{-2x^2/(\sum b_i^2)} dx$. ■

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

Lemma 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:*

$$\Pr\left(\left| d(v, U) - \frac{W_U}{t} \sum_{u \in T} \frac{d(v, u)}{w_u} \right| \leq \epsilon d(v, U) \right) \geq 1 - 2e^{-t\epsilon^2/8}$$

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

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 $c = 4d(v,U)/W_U$ (by Lemma 1 applied to $U \cup \{v\}$). Applying Lemma 4 and its corollary and scaling by W_U/t gives the lemma. ■

Lemma 6 *Let 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 $W_{U_L} \geq W_{U_R}$. Then, with probability at least $1 - \epsilon$, we have $|S \cap U_L| \geq 1/\epsilon^2$.*

Proof: Note that the probability that any fixed point of S falls in U_L is at least $1/2$ and that these events are independent. Thus, the probability distribution of t dominates the Binomial distribution $B(s, 1/2)$. The assertion of the lemma then follows from Lemma 4. ■

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

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 [FK98] and hybrid placement techniques of [GGR96].
The algorithm can be found in Figure 2. 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 1 *With probability at least $3/4$, the algorithm of Figure 2 computes a $(1 + O(\epsilon))$ approximation to Metric MIN-BISECTION. Its running time is $n^2 \cdot 2^{O(1/\epsilon^2)}$.*

3.1 Preliminary Properties

We start with the following Lemma.

Lemma 7 *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 $OPT + O(W\sqrt{\ell/n})$.*

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.
Using $X_u = 1\!\!1(u \in V_j)$, the cardinality of L_j^* can be written as $\sum_{u \in L^*} X_u$, and a Chernov bound shows that

$$E(|L_j^* - \frac{\epsilon}{2}|U||) = O(\sqrt{n/\ell}).$$

Similarly for R_j^*. Thus the expected number of points moved is $O(\sqrt{\ell n})$.
The change in value when going from (L^*, R^*) to (L, R) is at most the weight of the points which are moved. The points moved have random weights, hence the expected weight of the points moved equals $O(W\sqrt{\ell/n})$.
■

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, 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.**
$$\forall v \in V, \text{ let } e_v = \min\{\frac{\widehat{W_{U_L}}}{t} \sum_{u \in T} \frac{d(v,u)}{w_u} + d(v, B_L), w_v\}. \tag{1}$$

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 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
   $$f_v = \sum_{k<j} d(v, A_k) + \frac{\ell - (j-1)}{\ell} e_v, \tag{2}$$
   $$\widehat{b}(v) = f_v - (w_v - f_v).$$

   (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 $A = \cup_j A_j$ and $B = \cup_j B_j$.

7. **Output.** Output the best of the bisections (A, B) thus constructed.

Figure 2: A combinatorial algorithm for metric Minimum Bisection.

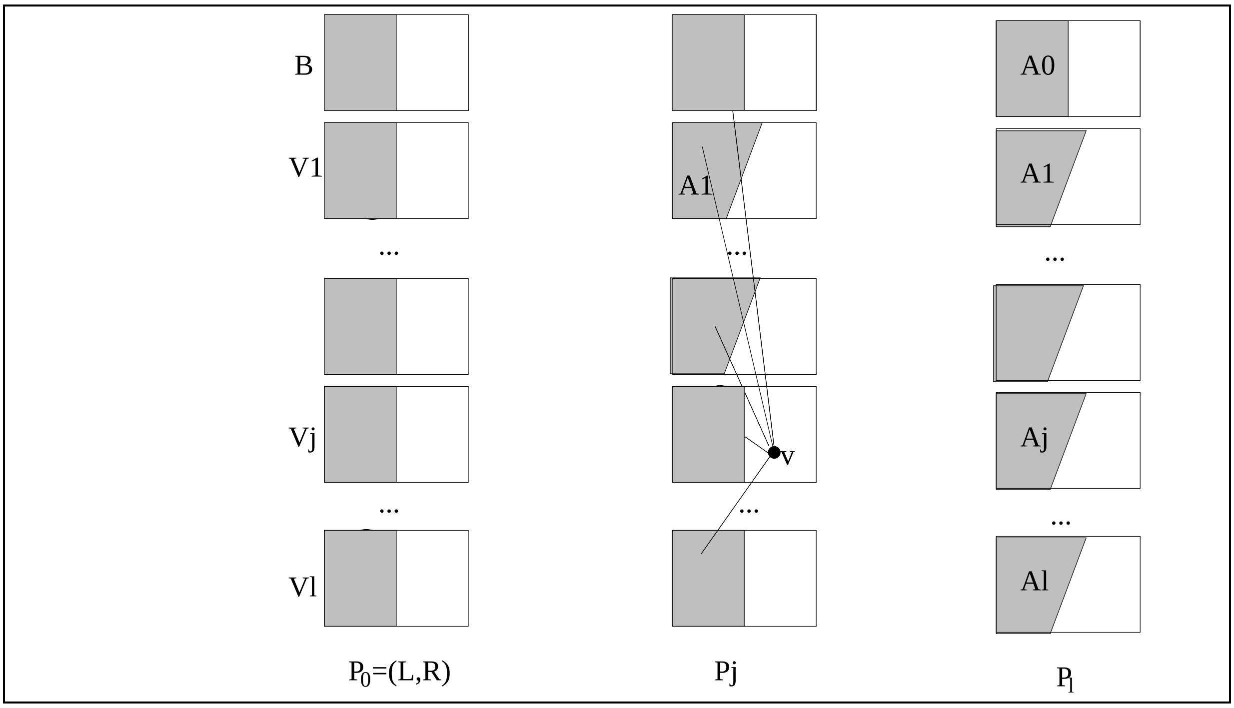

Figure 3: The hybrid partitions used by the combinatorial algorithm. f_v is an estimate of $d(v, \mathrm{Left}(P_j))$ for $v \in V_j$.

3.2 Proof of Theorem 1

The first part of the analysis is purely deterministic and, except for the last inequality, quite similar to the analysis in [GGR96].

3.2.1 Deterministic analysis

Let P_j be the following hybrid bisection:

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

The output is P_ℓ:

$$\mathrm{COST}(P_\ell) - \mathrm{COST}(P_0) \leq \sum_{1\leq j\leq \ell} [\mathrm{COST}(P_j) - \mathrm{COST}(P_{j-1})].$$

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 $b(u) = d(u, \mathrm{Left}(P_{j-1})) - (w_u - d(u, \mathrm{Left}(P_{j-1})))$. We have:

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

Now, here is the central part of the proof:

$$b(x_i) - b(y_i) = (b(x_i) - \widehat{b}(x_i)) + (\widehat{b}(x_i) - \widehat{b}(y_i)) + (\widehat{b}(y_i) - b(y_i)) \leq (b(x_i) - \widehat{b}(x_i)) + (\widehat{b}(y_i) - b(y_i)),$$

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

$$\begin{aligned} \text{COST}(P_j) - \text{COST}(P_{j-1}) &\leq \sum_{u \in V_j} |b(u) - \widehat{b}(u)| + 2d(V_j, V_j) && (3) \\ &\leq 2 \sum_{u \in V_j} |\sum_{k \geq j} d(u, L_k) - \frac{\ell - (j-1)}{\ell}(e_u - d(u, B_L))| + 2d(V_j, V_j). && (4) \end{aligned}$$

Now,

$$|\sum_{k \geq j} d(u, L_k) - \frac{\ell - (j-1)}{\ell}(e_u - d(u, B_L))| \leq$$

$$|\sum_{k \geq j} d(u, L_k) - \frac{\ell - (j-1)}{\ell} d(u, U_L)| + \frac{\ell - (j-1)}{\ell} |d(u, U_L) - (e_u - d(u, B_L))|. \quad (5)$$

We must now use probabilistic tools to analyze this equations.

3.2.2 Probabilistic analysis

We first analyze the first term of the right hand side of Equation 5.
Fix $v \in V_j$ and let $Z_v = \sum_{k \geq j} d(v, L_k)$. The expectation of Z_v is $d(v, U_L)$, 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 the sum into two parts, writing $Z_v = A_v + B_v$, 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 straighforward: applying Lemma 4 to A_v, with $b_i = w_v \epsilon / \sqrt{n}$, yields

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

For the second part, from Proposition 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 $d(u, v) \leq (w_u + w_v)/n \leq 2 w_u / n$. Applying Lemma 4 to B_v, with $b_u = 2 w_u / n$, now yields

$$E(|B_v - EB_v|) \leq \sqrt{\frac{\pi}{2} \frac{\sum_u 4 w_u^2}{n^2}}.$$

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

$$E(|B_v - EB_v|) \leq \sqrt{2\pi} \frac{\epsilon W}{n}.$$

Summing gives

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

As for the second term of Equation 5, from Lemma 5 applied to U_L, we have:

$$E(|d(v, U_L) - (e_u - d(u, B_L))|) \leq 2 \sqrt{\frac{2\pi}{t}} d(v, U_L) \leq 2 \sqrt{\frac{2\pi}{t}} w_v.$$

The rest of the proof is easy and entirely deterministic again.

3.2.3 Deterministic analysis

Plugging these bounds into Equation 4, we obtain:

$$E(\mathrm{COST}(P_j) - \mathrm{COST}(P_{j-1})) \leq 2 \sum_{u \in V_j} (\epsilon w_u + \sqrt{2\pi}\epsilon \frac{W}{n} + 2\sqrt{\frac{2\pi}{t}} w_u) + 2E(d(V_j, V_j)).$$

Summing over j, we get:

$$E(\mathrm{COST}(P_\ell) - \mathrm{COST}(P_0)) \leq 2[\epsilon W + \sqrt{2\pi}\epsilon W + 2\sqrt{\frac{2\pi}{t}} W] + 2E(\sum_j d(V_j, V_j)).$$

The last term is easy to deal with: its expectation is bounded by W/ℓ.
From Lemma 6, $t \geq 1/\epsilon^2$ with probability at least $1 - \epsilon$, and then, with Lemma 7 we obtain:

$$E(\mathrm{COST}(P_\ell) - \mathrm{OPT}) \leq 2W[\epsilon(1 + 3\sqrt{2\pi}) + \frac{1}{\ell} + O(\sqrt{\frac{\ell}{n}})].$$

Using Markov's inequality, remembering that $\ell = 1/\epsilon$ and comparing with the lower bound from Lemma 2 then concludes the proof of the Theorem.

■

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 $\ell \in [1/\epsilon, n\epsilon^2]$. Indeed, going back to previous work on dense graphs, one may have been intrigued to notice that a partition into $\ell = 1/\epsilon$ parts was used in [GGR96], while a partition into $\ell = n\epsilon^2$ parts was used in [F96]. Indeed, we now see from the above analysis that, with our algorithm, the number of parts is largely irrelevant: this may serve as an explanation. Perhaps the algorithm is nicer to think about in the case when $\ell = n\epsilon^2$, since it is then very close to a natural greedy algorithm: take the vertices by groups of $1/\epsilon^2$ at a time, and bisect each group in the best possible way, taking into account the choices made so far (and adding an estimate to take into account the vertices not yet considered.)

2. The running time can be improved in a manner similar to [GGR96]: first, in Equation 2, instead of calculating $d(v, A_i)$ exactly, we could estimate it via sampling, thus gaining a factor of n. Second, instead of running the algorithm on the whole graph, we could run it on a (larger) sample of the point set.

3. Except for biased sampling, which is specific to the metric situation, the additional ideas used here to modify the hybrid placement technique from [GGR96] can be applied to the dense graphs setting as well. We conjecture that in dense graphs, it might be possible to use ideas from our combinatorial algorithm so as to improve the query complexity from [GGR96] by a factor of $O(1/\epsilon)$.

4. Considering the metric versus dense graph settings, let us compare our combinatorial algorithm in the metric setting with its natural analog in the dense graph setting:

   - In the metric setting, some vertices can have overwhelming importance (the ones with weight close to W), and so we need to set those vertices aside and treat them separately. This does not 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 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; instead the terms in the sums can be quite large; thus a more sophisticated version of Hoeffding's inequality is required, and applying it requires a much more delicate analysis.
- 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.

5. Focusing on dense graphs, let us compare the dense graph analog of our combinatorial algorithm to the combinatorial algorithm from [GGR96]:
   - The hybrid placement technique, which was introduced in [GGR96], is central to our algorithm as well.
   - We sample $O(1/\epsilon^2)$ points in total, as opposed to $\Omega(1/\epsilon^3 \ln(1/\epsilon))$.
   - The partition (V_j) is random instead of arbitrary (necessary for this smaller sampling to work).
   - Our estimator is slightly different, since we do not re-sample the hybrid partitions, but instead use an estimator which combines the distances to vertices already classified with a scaled version of the original estimate. This is necessary for the smaller sampling to work.
   - For partitioning into two parts, we only use sampling to estimate for the distance from v to the left side of the partition; since the sum of its distances to the left and to the right side is equal to its degree, this immediately implies an estimate for the distance from v to the right side of the partition. (This is a detail).
   - In the analysis, instead of separating the point set into "normal" and "exceptional" vertices, we just integrate Hoeffding's formula so as to directly use the resulting formula for the expected deviation from the mean. (It would however still have been possible to prove the result with a separation into normal and exceptional vertices).

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 [AKK95]. 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 [AKK95].)
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 11), we solve by exhaustive search. Otherwise, we run the algorithm presented on Figure 4 at the end of the paper. Throught this section we will refer to the notation used in the description of this algorithm.

Theorem 2 *With probability at least $3/4$, the algorithm in Figure 4 computes a $(1+6(60\sqrt{2\pi}+3)\epsilon)$ 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.*

4.1 Proof of Theorem 2

Let (x_v) be the optimal bisection, (x_v^*, y_v^*) the optimal fractional solution of the linear program, (y_v) the partition obtained by the randomized rounding, and (y'_v) the bisection output by the algorithm.

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.**
$$\forall v \in V, \text{ let } e_v = \min\{\frac{\widehat{W_{U_L}}}{t} \sum_{u \in T} \frac{d(v, u)}{w_u} + d(v, B_L), w_v\}. \tag{6}$$

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.
$$\begin{cases} \forall v, \;\; 0 \leq x_v & \leq \;\; 1 \\ \forall v, \;\; d(v, B_L) + \sum_{u \in U} (1 - x_u) d(u, v) & \leq \;\; e_v + z_v \\ \forall v, \;\; z_v & \geq 0 \\ \quad \sum_v z_v & \leq \;\; 20\sqrt{2\pi}\epsilon W \\ \quad \sum_v x_v + |B_L| & = \;\; n/2. \end{cases}$$
   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 4: A linear programming algorithm for metric Minimum Bisection.

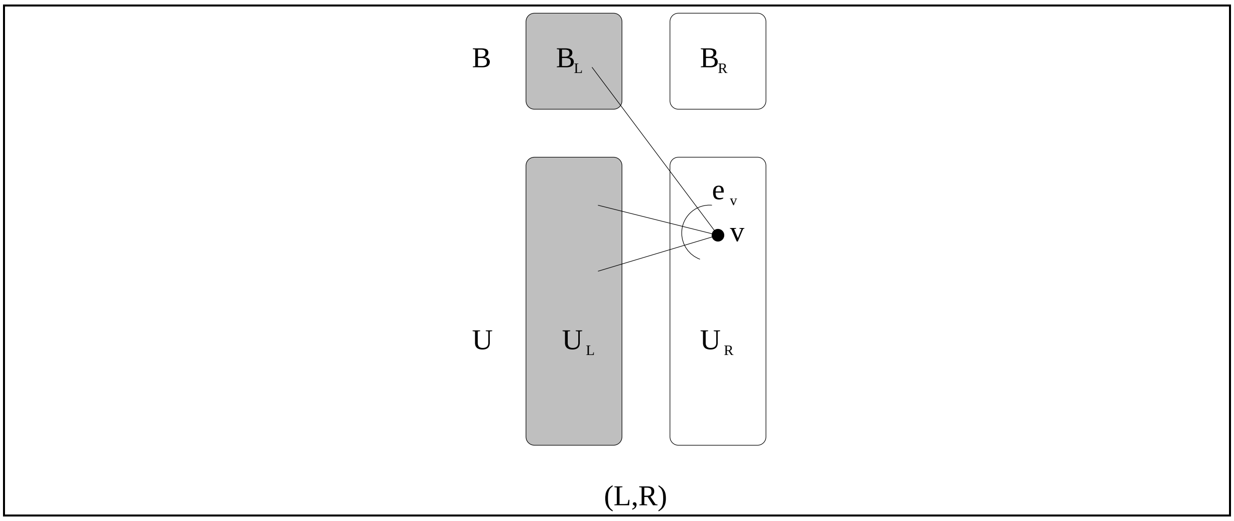

Figure 5: The partition used by the linear programming algorithm. e_v is an estimate of $d(v, L) = d(v, U_L) + d(v, B_L)$.

Lemma 8 *With probability at least* $89/100$ *we have that the optimum solution* (x_v) *is feasible and moreover that*

$$OPT = \mathrm{COST}(x_v) \geq \mathrm{COST}(x_v^*) - 20\sqrt{2\pi}\epsilon W.$$

Proof: Let δ_v be the difference between e_v and its expectation. By Lemma 5, we have that

$$\mathbf{E}\left(\sum_U |\delta_v|\right) \leq 2\sqrt{\frac{2\pi}{t}}W.$$

Using Lemma 6, we can assume that $t \geq 1/\epsilon^2$, and use Markov's Inequality to get that

$$\Pr\left(\sum_U |\delta_v| \leq 20\sqrt{2\pi}\epsilon W\right) \geq 9/10$$

for sufficiently small ϵ. This shows the feasibility of (x_v) with probability $89/100$ and proves also the second part of the lemma since $\mathrm{COST}(x_v)$ differs from $\mathrm{COST}(x_v^*)$ by at most $\sum_U |\delta_v|$. ■

Lemma 9 *With probability at least* $1 - 1/100$*, we have:* $c(x^*) + 2\epsilon W \geq c(y)$.

Proof: We must bound above the sum $S = \sum_U z_v a_v$, where $z_v = x_v^* - y_v$, and $a_v = e_v - d(v, B_R)$ $v \in U$. Note that the absolute values of the a_v are all bounded above by $\epsilon^2 W/100$. Since their sum is at most W we have that the variance of S is bounded above $\epsilon^2 W^2/100$. Using Chebychev's inequality we get that S is bounded above by $\epsilon W/10$ with probability $1 - 1/100$. ■

Lemma 10 *With probability at least* $1 - 1/10$*, we have* $c(y) + 40\sqrt{2\pi}\epsilon W \geq \mathrm{COST}(y_v)$.

Proof: We have, with probability $1 - 1/10$,

$$\begin{aligned} |c(y) - \mathrm{COST}(y_v)| &= |\sum_U y_v \left(d(v, L) - e_v - z_v^*\right)| \\ &\leq \sum_U z_v^* + \sum_U |d(v, L) - e_v| \\ &\leq 20\sqrt{2\pi}\epsilon W + 20\sqrt{2\pi}\epsilon W \end{aligned}$$

from the LP and from Lemma 5 applied to L, followed by Markov's inequality. ■

Lemma 11 *With probability at least* $1 - 1/100$*, we have* $\mathrm{COST}(y'_v) \leq \mathrm{COST}(y_v) + \epsilon W$.

Proof: Note that the y_v have expectation x^*_v and variance bounded above by $1/4$. The sum $Z = \sum_V y_v$ has expectation $n/2$ and variance at most $n/4$. Chebychev's Inequality gives us that

$$\Pr(|Z - n/2| \leq \epsilon n) \geq 1 - 4\epsilon^2 \geq 1 - 1/100$$

for sufficiently small ϵ. The lemma follows now from the fact that the sum of the ϵn smallest weights does not exceed ϵW. ■

To prove Theorem 2, it now suffices to combine Lemmas 11, 10, 9 and 8 so as to prove that the value of the partition output is at most $\mathrm{OPT} + O(\epsilon W)$. By Lemma 2, this is at most $(1 + O(\epsilon))\mathrm{OPT}$.

The running time follows by inspection. ■

Remarks.

1. Except for biased sampling, which is specific to the metric situation, the additional ideas used here to modify the algorithm from [AKK95] can be applied to the dense graphs setting as well.

2. Considering the metric versus dense graph settings, let us compare our combinatorial algorithm in the metric setting with its natural analog in the dense graph setting:

   - In the metric setting, some vertices can have overwhelming importance (the ones with weight close to W), and so we need to set those vertices aside and treat them separately. This does not 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 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 analysis no longer deals with sums of $\{0, 1\}$ variables; instead the terms in the sums can be quite large; thus a more sophisticated version of Hoeffding's inequality is required, and applying it requires a much more delicate analysis.
   - 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.

3. Focusing on dense graphs, let us compare the dense graph analog of our combinatorial algorithm to the combinatorial algorithm from [AKK95]:

   - The linearized programming technique, which was introduced in [AKK95], is central to our algorithm as well.
   - We sample $O(1/\epsilon^2)$ points in total, as opposed to $\Omega(1/\log n)$.
   - We modify the LP slightly by introducing n new variables z_v, to make the constraints more flexible. This is necessary for the smaller sample to work.

5 Metric MAX-CUT Revisited

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 [FK98].

Theorem 3 *There is a PTAS for Metric MIN-BISECTION, with running time is* $O(n^2 \cdot 2^{O(1/\epsilon^2)})$.

6 Extensions

6.1 Extension to $(k, n-k)$ Metric MIN-PARTITIONING

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

Proof: There are two cases according to the values of the ratio k/n and of the accuracy requirement ϵ.
(i) Suppose first that $k/n \geq \epsilon/2$. Then we apply one of the above algorithms, say the second one, with $\epsilon' = \epsilon^2$ and the necessary modifications concerning the sizes constraints: we run two distinct LPs, one with $|L| = k$ and the other one with $|L| = n-k$. This ensures that in one of these programs we have $W_{U_L} \geq W_{U_R}$.
(ii) Suppose now that $k/n < \epsilon/2$. We claim that in this case a solution with approximation ratio $1+\epsilon$ is obtained just by separating the k points with smallest weights from the rest. In order to prove this claim, fix attention first on 2 vertices x_1, x_2. Let w_i be the weight of x_i. For any other vertex x_3 we have of course

$$d(x_1, x_2) \leq d(x_1, x_3) + d(x_3, x_2)$$

Summing over all choices for x_3, this gives:

$$w_1 + w_2 \geq n d(x_1, x_2)$$

Take now k vertices $x_1, x_2, \ldots x_k$. The preceding inequality gives

$$(k-1)\sum_1^k w_i \geq n \sum_{i<j} d(x_i, x_j) \tag{7}$$

Let $U \subseteq V$. The value of the partition $(U, V \backslash U)$ is

$$\mathrm{Val}(U, V\backslash U) = \sum_{x_i \in U} w_i \; - 2 \sum_{x_i, x_j \in U} d(x_i, x_j)$$

Thus,

$$\begin{aligned} \mathrm{OPT} \quad &\geq \quad \min_{|S|=k} \left(\sum_{x_i \in S} w_i - 2 \sum_{x_i, x_j \in S} d(x_i, x_j) \right) \\ &\geq \quad \left(1 - \frac{2(k-1)}{n}\right) \min_{|S|=k} \sum_{x_i \in S} w_i, \end{aligned}$$

the last by using equation (7). ∎

6.2 Extension to Size Constraint Metric MIN-PARTITIONING

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

Proof: We use the following extension of our linear programming algorithm for $(k, n-k)$ MIN-PARTITIONING.

1. If n is less than a certain constant, use exhaustive search. Otherwise do the following.
2. Let n_1 be the largest size. If $n - n_1 \leq \epsilon n$, then use the $(k, n-k)$ MIN-PARTITIONING algorithm of section 5.1 with $k = n - n_1$. Then partition the smallest part arbitrarily into parts of sizes $n_2, n_3, ..n_K$. If $n - n_1 > \epsilon n$, do the following
3. The weight W_X of a part X will be called *large* if it exceeds $\epsilon W/(K-1)$. Let h be the number of parts with large weight in an optimum solution, and let $n_1, n_2, \ldots, n_h$ denote their sizes. We solve in what follows the partitioning $(n_1, n_2, \ldots, n_h, n - \sum n_i)$ on (V, d).
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 in the value of s comparatively to its value of s in algorithm of figure 2. This is due to the fact that now the lower bound of OPT that we have is only $\Omega(\epsilon W)$ instead of $\Omega(W)$ for the MIN-BISECTION algorithm.
6. 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. Also guess the approximate weight $\tilde{W}_i$ of that part. Note that the number of samples needed for a correct guess has order $n^{O(1/\epsilon^2)}$.
7. For each $v \in U$ and for each i, let
$$e_{v,i} = \min\{\frac{\tilde{W}_i}{t_i} \sum_{u \in T_i} \frac{d(u,v)}{w_u} + d(v, B_i), w_v\}.$$
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:
$$\min c(x)$$
subject to the constraints
$$\begin{cases} \forall v, i \;\; x_{v,i} & \geq & 0 \\ \forall v, \;\; \sum_i x_{v,i} & = & 1 \\ \forall v, i \;\; d(v, B_i) + \sum_{u \in U} (x_{u,i}) d(u, v) & \leq & e_{v,i} + z_{v,i} \\ \forall v, i \;\; z_{v,i} & \geq 0 & \\ \qquad \sum_i \sum_v z_{v,i} & \leq & 3\epsilon W \\ \forall i, \;\; |B_i| + \sum_{v \in U} x_{v,i} & = & n_i \end{cases}$$
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 of V.
10. Repair sizes analogously to the last step of the linear programming MIN-BISECTION algorithm.

This ends the description of the algorithm. We prove in what follows the correctness of the above algorithm. A key observation is the following. With a partition $A_1, A_2, ...A_K$ with part sizes $n_1, n_2, ...n_K$ we associate the $(n, n - n_1)$ partition (A_1, B) whhith $B = A_2 \cup A_3 ... \cup A_K$. By Lemma 3 we have that the value of this partition is at least
$$W \frac{n_1(n - n_1)}{2((n-1)(n-n_1) + n_1(n_1 - 1))}.$$
We distinguish between two cases (i) and (ii):
Case (i) If $n - n_1 \leq \epsilon n$, then the correctness follows from the correctness of the $(k, n-k)$ MIN-PARTITIONING algorithm,

Case (ii) In this case, the above formula gives us that the value of the partition (A_1, B) is at least

$$\frac{W(1-\epsilon).\epsilon n}{2((n^2(1-\epsilon)+\epsilon^2 n^2)} \geq \frac{\epsilon W}{3}$$

Plainly, this lower bound is also valid for the optimum of the problem. Our algorithm gives in this case an additive approximation $O(\epsilon^2 W)$, which by what as just been proved guarantees an approximation ratio $1+O(\epsilon)$ as desired. This ends the proof of Theorem 5.

■

7 Further research

An interesting open problem is to improve running times of our PTASs as well as their sample complexity (also in the sense of random "sub-problem" sample complexity of [AFKK02]). Our Linear Program PTAS is based on an extension of the notion of a smooth polynomial program (cf. [AKK95]). An interesting open problem is how far such an extension can be carried out. Another question would be to shed some light on the size-constraint (in the general sense of this paper) MIN-SUM-K-CLUSTERING problems (cf. [FKKR03]).

Acknowledgements. We thank Mark Jerrum, Uri Feige, Alan Frieze, and Ravi Kannan for stimulating discussions.

References

[AFKK02] 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.

[AKK95] 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.

[B62] G. Bennet,*Probability inequalities for sums of independent random variables*, Journal of the American Statistical Association 57 (1962) 33-45

[BK01] 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.

[F02] U. Feige, Relations between Average Case Complexity and Approximation Complexity, Proc. 34^{th} ACM STOC (2002), pp. 534-543.

[FK00] U. Feige and R. Krauthgamer, *A Polylogarithmic Approximation of the Minimum Bisection*, Proc. 41st IEEE FOCS (2000), pp. 105–115.

[F96] W. Fernandez de la Vega, *MAX-CUT has a Randomized Approximation Scheme in Dense Graphs,* Random Structures and Algorithms 8 (1996), pp.187-198

[FVK00] 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.

[FK98] 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.

[FKKR03] W. Fernandez de la Vega, M.Karpinski, C. Kenyon, and Y. Rabani, *Approximation schemes for clustering problems*, Proc. 35th ACM STOC(2003), to appear.

[FK96] A.M. Frieze and R. Kannan, *The Regularity Lemma and Approximation Schemes for Dense Problems,* Proc. 37th FOCS (1996), pp. 12-20.

[FK97] A.M. Frieze and R. Kannan, *Quick Approximation to Matrices and Applications,* Combinatorica 19 (2) (1999), pp. 175-120.

[GGR96] 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.

[HM98] M. Habib, C. McDiarmid, J. Ramirez-Alfonsin, B. Reed (eds.), *Probabilistic Methods for Algorithmic Discrete Mathematics*, Concentration, Springer, 1998.

[H97] D.S. Hochbaum(ed.), Approximation Algorithms for NP-Hard Problems, PWS, 1997.

[H63] W. Hoeffding, *Probability inequalities for sums of bounded random variables*, Journal of the American Statistical Association 58 (1963) 13-30.

[I99] P. Indyk, *A Sublinear Time Approximation Scheme for Clustering in Metric Spaces,* Proc. 40th IEEE FOCS (1999), 154-159.

[K02] M. Karpinski, Approximability of the Minimum Bisection Problem, Proc. 27^{th} MFCS (2002), LNCS Vol. 2420, Springer, 2002, pp. 59-67.

[MCD98] C. McDiarmid, *Concentration*, Probabilistic Methods for Algorithmic Disctrete Mathematics, M.Habib, C.McDiarmid, J. Ramirez Alfonsin, B.Reed (Eds.), Springer 1998.

[PY91] C. H. Papadimitriou and M. Yannakakis. *Optimization, Approximation and Complexity Classes*, J. Comput. System Sciences, 43 (1991), pp. 425-440.

[S00] L.J. Schulman. Clustering for edge-cost minimization. Proc. 32nd ACM STOC (2000), pp. 547–555.