This is a summary of the talk I gave on November 9th of 2011 in University of Waterloo. The aim is to describe one of the main results of the paper

> Min-Max Graph Partitioning and Small Set Expansion by Bansal, Feige, Krauthgamer, Makarychev, Nagarajan, Naor and Schwartz (FOCS'11).

The version available on http://arxiv.org/abs/1110.4319v2 was used.
Abbas Mehrabian

Problem (Small Set Expansion). Given edge-weighted graph G and $\rho \in (0, 0.5]$, find

$$\text{OPT} := \min_{|S| \leq \rho n} \frac{\delta(S)}{|S|}.$$

An $O(\log n)$ approximation can be derived using Räcke'08.

The authors prove the following theorem in the paper, which we will not prove, but shall use a corollary of it.

Theorem 1. For every fixed $\epsilon > 0$ there is an algorithm that outputs a set S of size $\leq (1+\epsilon)\rho n$ with edge expansion $O\left(\sqrt{\log n \log(1/\rho)}\right)$ OPT.

Remarks.

1. This algorithm and all following algorithms are randomized, have polynomial expected running time, and produce the desired output with high probability.

2. The algorithm uses an SDP relaxation, and uses "orthogonal separators," (introduced by Chlamtac, Makarychev, Makarychev'06) for rounding it.

3. A more general theorem is proved, in which the vertices are also weighted, and there is a lower bound on the weight of the set S.

4. For graphs excluding a fixed minor/having fixed genus, the approximation factor is improved to $O(1)$. The proof uses an LP relaxation and a new notion called "LP separators," and the authors build these using "separating decompositions" of graph metrics.

Problem (Weighted ρ-Unbalanced Cut). Given graph G, vertex weights y, edge weights w and $\rho \in (0, 1]$ the goal is to find S minimizing $\delta(S)$ satisfying

$$y(S) \geq \rho y(V) \qquad \text{and} \qquad |S| \leq \rho n$$

Corollary 1. For every fixed $\epsilon > 0$ there is an algorithm that finds set S with $|S| \leq \beta \rho n$, $y(S) \geq \rho y(V)/\gamma$ and $\delta(S) \leq \alpha\,\text{OPT}$, where $\alpha = O\left(\sqrt{\log n \log(1/\rho)}\right)$, $\beta = 1 + \epsilon$, and $\gamma = O(1)$.

Problem (Min-Max k-Partitioning). Given edge-weighted graph G and positive k that divides n, partition $V(G)$ into $S_1, \ldots, S_k$ of equal size so as to minimize $\max \delta(S_i)$.

Best known algorithm based on previous work is a (true) $O(k\sqrt{\log n})$-approximation.

We will prove the following.

Theorem 2. For every fixed $\epsilon > 0$ there is an algorithm that outputs $S_1, \ldots, S_k$ with $\max |S_i| \leq 2(1+\epsilon)n/k$ and $\max \delta(S_i) \leq O\left(\sqrt{\log n \log k}\right)$ OPT.

(The following example shows that greedily using Corollary 1 might give a solution as bad as $\Omega(k)$ OPT.

$$\vdots$$
$$\vdots$$

)

The algorithm has two phases. In the first phase (`Algorithm 1`) using Corollary 1 as a procedure we generate a family $\mathcal{S}$ of subsets of V of size $\leq \beta n/k$ (note that $\mathcal{S}$ is a multiset, i.e. may contain several copies of the same subset), whose every element has small expansion, and also has a certain "uniformity" constraint. In the second phase (`Algorithm 2`) we generate the partition $S_1, S_2, \ldots, S_k$ using this family.

`Algorithm 1` is shown in the next page (taken from the original paper).

Let $\mathcal{C} = \{S \subseteq V : |S| \le n/k\}$ denote all the vertex-sets that are feasible for a single part. Note that a feasible solution in Min–Max k–Partitioning corresponds to a partition of V into k parts, where each part belongs to $\mathcal{C}$. Algorithm 1, described below, *uniformly covers* V using sets in $\mathcal{C}$ (actually a slightly larger family than $\mathcal{C}$). It is important to note that its output $\mathcal{S}$ is a multiset.

Algorithm 1: Covering Procedure for Min–Max k–Partitioning:

```
Set t = 1, and y^1(v) = 1 for all v ∈ V
while Σ_{v∈V} y^t(v) > 1/n do
    // Solve the following using algorithm from Corollary 2.7.
    Let S^t ⊆ V be the solution for Weighted ρ-Unbalanced Cut instance ⟨G, y^t, w, 1/k, 1/k⟩.
    Set 𝒮 = 𝒮 ∪ {S^t}.
    // Update the weights of the covered vertices.
    for every v ∈ V do
        Set y^{t+1}(v) = 1/2 · y^t(v) if v ∈ S^t, and y^{t+1}(v) = y^t(v) otherwise.
    Set t = t + 1.
return 𝒮
```

Theorem 3.1. *Running Algorithm 1 on an instance of Min–Max k–Partitioning outputs $\mathcal{S}$ that satisfies (here* OPT *denotes the optimal value of the instance):*

1. *For all $S \in \mathcal{S}$ we have $\delta(S) \le \alpha \cdot \mathsf{OPT}$ and $|S| \le \beta \cdot n/k$.*
2. *For all $v \in V$ we have $|\{S \in \mathcal{S} : S \ni v\}|/|\mathcal{S}| \ge 1/(5\gamma k)$.*

Proof. For an iteration t, let us denote $Y^t := \sum_{v\in V} y^t(v)$. The first assertion of the theorem is immediate from the following claim.

Claim 3.2. *Every iteration t of Algorithm 1 satisfies $\delta(S^t) \le \alpha \cdot \mathsf{OPT}$ and $|S^t| \le \beta \cdot n/k$.*

Proof. It suffices to show that the optimal value of the Weighted ρ-Unbalanced Cut instance $\langle G, y^t, w, \frac{1}{k}, \frac{1}{k}\rangle$ is at most OPT. To see this, consider the optimal solution $\{S_i^*\}_{i=1}^k$ of the original Min–Max k–Partitioning instance. We have $|S_i^*| \le n/k$ and $w(\delta(S_i^*)) \le \mathsf{OPT}$ for all $i \in [k]$. Since $\{S_i^*\}_{i=1}^k$ partitions V, there is some $j \in [k]$ with $y^t(S_j^*) \ge Y^t/k$. It now follows that S_j^* is a feasible solution to the Weighted ρ-Unbalanced Cut instance $\langle G, y^t, w, \frac{1}{k}, \frac{1}{k}\rangle$, with objective value at most OPT, which proves the claim. $\square$

We proceed to prove the second assertion of Theorem 3.1. Let ℓ denote the number of iterations of the while loop, for the given Min–Max k–Partitioning instance. For any $v \in V$, let N_v denote the number of iterations t with $S^t \ni v$. Then, by the y-updates we have $y^{\ell+1}(v) = 1/2^{N_v}$. Moreover, the termination condition implies that $y^{\ell+1}(v) \le 1/n$ (since $Y^{\ell+1} \le 1/n$). Thus we obtain $N_v \ge \log_2 n$ for all $v \in V$. From the approximation guarantee of the Weighted ρ-Unbalanced Cut algorithm, it follows that $y^t(S^t) \ge \frac{1}{\gamma k} \cdot Y^t$ in every iteration t. Thus $Y^{t+1} = Y^t - \frac{1}{2} \cdot y^t(S^t) \le \left(1 - \frac{1}{2\gamma k}\right) \cdot Y^t$. This implies that $Y^\ell \le \left(1 - \frac{1}{2\gamma k}\right)^{\ell-1} \cdot Y^1 = \left(1 - \frac{1}{2\gamma k}\right)^{\ell-1} \cdot n$. However $Y^\ell > 1/n$ since the algorithm performs ℓ iterations. Thus, $\ell \le 1 + 4\gamma k \cdot \ln n \le 5\gamma k \cdot \log_2 n$. This proves $|\{S \in \mathcal{S} : S \ni v\}|/|\mathcal{S}| = N_v/\ell \ge (5\gamma)^{-1}k^{-1}$. $\square$

Lemma 1. Let α, β, γ be the same as Corollary 1.

1. For all $S \in \mathcal{S}$ we have $\delta(S) \leq \alpha\,\mathrm{OPT}$ and $|S| \leq \beta n/k$.

2. For all $v \in V$ we have $|\{S \in \mathcal{S} : v \in S\}|/|\mathcal{S}| \geq 1/5\gamma k$.

Proof. 1. Let $S_1^*, \ldots, S_k^*$ be the optimal solution for the original min-max graph partitioning problem and fix iteration index t. For some i we have $y^t(S_i^*) \geq y^t(V)/k$ hence S_i^* is feasible for weighted ρ-unbalanced cut with weights y^t. Done by Corollary 1.

2. Let ℓ be the total number of iterations, and N_v be number of times t with $v \in S^t$. By termination, for all v we have $y^{\ell+1}(v) \leq 1/n$ hence $N_v \geq \log_2 n$.

   From Corollary 1, $y^t(S^t) \geq y^t(V)/\gamma k$ for all t. Thus

   $$y^{t+1}(V) = y^t(V) - \frac{1}{2}y^t(S^t) \leq \left(1 - \frac{1}{2\gamma k}\right) y^t(V),$$

   so

   $$y^\ell(V) \leq \left(1 - \frac{1}{2\gamma k}\right)^{\ell-1} y^1(V),$$

   but $y^\ell(V) > 1/n$ (because the algorithm had ℓ iterations) and $y^1(V) = n$, so

   $$\ell \leq 1 + 4\gamma k \ln n \leq 5\gamma k \log_2 n,$$

   hence

   $$N_v/\ell \geq \log_2 n/5\gamma k \log_2 n = 1/5\gamma k. \quad \square$$

The following lemma will be used in the analysis, see the original paper for the proof.

The Aggregation Lemma. Let $a_1, \ldots, a_t, b_1, \ldots, b_t, A, B, S, T$ be nonnegative reals satisfying

$$a_i < A, b_i < B, \sum a_i \leq S, \sum b_i \leq T.$$

Assume that for all $i \neq j$, at least one of $a_i + a_j > A$ or $b_i + b_j > B$ holds. Then

$$t < \frac{S}{A} + \frac{T}{B} + \max\left\{\frac{S}{A}, \frac{T}{B}, 1\right\}.$$

`Algorithm 2`, shown in the next page (taken from the original paper) receives as input a family (multiset) $\mathcal{S}$ of subsets of V such that

i) all $S \in \mathcal{S}$ satisfy $|S| \leq 2n/k$ and $\delta(S) \leq B$, and

ii) every $v \in V$ is covered by at least a c/k fraction of the members of $\mathcal{S}$ (for some $c \in (0, 1)$;

and outputs a partition of V.

Algorithm 2: Aggregation Procedure for Min–Max k–Partitioning:

1 `Sampling`
   Sort sets in $\mathcal{S}$ in a random order: $S_1, S_2, \dots, S_{|\mathcal{S}|}$. Let $P_i = S_i \setminus \cup_{j<i} S_j$.

2 `Replacing Expanding Sets with Sets from` $\mathcal{S}$
   while *there is a set* P_i *such that* $\delta(P_i) > 2B$ **do**
      Set $P_i = S_i$, and for all $j \neq i$, set $P_j = P_j \setminus S_i$.

3 `Aggregating`
   Let $B' = \max\{\frac{1}{k}\sum_i \delta(P), 2B\}$.
   while *there are* $P_i \neq \varnothing$, $P_j \neq \varnothing$ $(i \neq j)$ *such that* $|P_i| + |P_j| \leq 2(1+\varepsilon)n/k$ *and* $\delta(P_i) + \delta(P_j) \leq 2B'\varepsilon^{-1}$ **do**
      Set $P_i = P_i \cup P_j$ and set $P_j = \varnothing$.

4 **return** all non-empty sets P_i.

$v \notin S$), $\Pr((u,v) \in E(P_i, \cup_{j>i} P_j) \mid S_i = S) \leq \Pr(v \notin \cup_{j<i} S_j \mid S_i = S) \leq (1 - c/k)^{i-1}$, since v is covered by at least c/k fraction of sets in $\mathcal{S}$ and is not covered by $S_i = S$. Hence,

$$\mathbb{E}[w(E(P_i, \cup_{j>i} P_j)) \mid S_i = S] \leq (1-c/k)^{i-1}\delta(S) \leq (1-c/k)^{i-1} B,$$

and $\mathbb{E}[w(E(P_i, \cup_{j>i} P_j))] \leq (1-c/k)^{i-1} B$. Therefore, the total expected weight of edges crossing the boundary of P_i's is at most $\sum_{i=0}^{\infty} (1-c/k)^i B = kB/c$, and $\mathbb{E}\big[\sum_i \delta(P_i)\big] \leq 2kB/c$.

2. After each iteration of step 2, the following invariant holds: the collection of sets $\{P_i\}$ is a partition of V and $P_i \subset S_i$ for all i. Particularly, $|P_i| \leq |S_i| \leq 2n/k$. The key observation is that at every iteration of the "while" loop, the sum $\sum_j \delta(P_j)$ decreases by at least $2B$. This is due to the following uncrossing argument:

$$\begin{aligned}
\delta(S_i) + \sum_{j \neq i} \delta(P_j \setminus S_i) &\leq \delta(S_i) + \sum_{j \neq i} \Big(\delta(P_j) + w(E(P_j \setminus S_i, S_i)) - w(E(S_i \setminus P_j, P_j)\Big) \\
&\leq \delta(S_i) + \Big(\sum_{j \neq i} \delta(P_j)\Big) + \underbrace{w(E(V \setminus S_i, S_i))}_{\delta(S_i)} - \underbrace{w(E(P_i, V \setminus P_i))}_{\delta(P_i)} \\
&= \Big(\sum_j \delta(P_j)\Big) + 2\delta(S_i) - 2\delta(P_i) \leq \Big(\sum_j \delta(P_j)\Big) - 2B.
\end{aligned}$$

we used that $P_i \subset S_i$, all P_j are disjoint, $\cup_{j \neq i}(P_j \setminus S_i) \subset V \setminus S_i$, $P_i \subset S_i \setminus P_j$, $\cup_{j \neq i} P_j = V \setminus P_i$. Hence, the number of iterations of the loop in step 2 is always polynomially bounded and after the last iteration $\mathbb{E}\big[\sum_i \delta(P_i)\big] \leq 2kB/c$ (the expectation is over random choices at step 1; the step 2 does not use random bits). Hence, $\mathbb{E}[B'] \leq 4B/c$.

3. The following analysis holds conditional on any value of B'. After each iteration of step 3, the following invariant holds: the collection of sets $\{P_i\}$ is a partition of V. Moreover, $|P_i| \leq 2(1+\varepsilon)n/k$ and $\delta(P_i) \leq 2B'\varepsilon^{-1}$ (note: after step 2, $\delta(P_i) \leq 2B \leq B'$ for each i).

When the loop terminates, we obtain a partition of V into sets P_i satisfying $|P_i| \leq 2(1+\varepsilon)n/k$, $\sum_i |P_i| = n$, $\delta(P_i) \leq 2B'\varepsilon^{-1}$, $\sum_i \delta(P_i) \leq kB'$, such that no two sets can be merged without violating above constraints. Hence by Lemma 3.4 below (with $a_i = |P_i|$ and $b_i = \delta(P_i)$), the number of non-empty sets is at most $2\,\frac{n}{2(1+\varepsilon)n/k} + \frac{kB'}{2B'\varepsilon^{-1}} = (1+\varepsilon)^{-1}k + (\varepsilon/2)k \leq k$. ∎

Lemma 2. For every fixed $\epsilon \in (0,1)$, `Algorithm 2` outputs a partition $\mathcal{P}$ of $V(G)$ into at most k sets such that

$$\mathbb{E}[\max \delta(P) : P \in \mathcal{P}] \leq 4B/c\epsilon,$$

and $|P| \leq 2(1+\epsilon)n/k$ for all $P \in \mathcal{P}$.

Proof. Let $\Delta := \sum \delta(P_i)$. ($\Delta/2$ is the number of crossing edges.) Notice that during the algorithm the P_i's always partition $V(G)$, and for all i we have $|P_i| \leq 2(1+\epsilon)n/k$, and during the first two steps $P_i \subseteq S_i$.

Claim 1. After step 1 (`Sampling`) finishes, $\mathbb{E}[\Delta] \leq 2kB/c$.

Proof.

$$\Delta/2 = \sum_i w(E(P_i, \cup_{j>i} P_j))$$

Fix some i and some $S \in \mathcal{S}$ and we bound $\mathbb{E}\left[w(E(P_i, \cup_{j>i} P_j))\right]$ conditioned on $S_i = S$. Fix uv with $u \in P_i, v \in \cup_{j>i} P_j$ so $u \in S, v \notin S$. If $uv \in E(P_i, \cup_{j>i} P_j)$ then v was not covered in the first $i-1$ sets in the chosen order. However, v is covered by a c/k fraction of the sets in $\mathcal{S}$, hence the probability of this is at most $(1-c/k)^{i-1}$, thus

$$\mathbb{E}[w(E(P_i, \cup_{j>i} P_j)) | S_i = S] \leq (1-c/k)^{i-1}\delta(S) \leq (1-c/k)^{i-1}B,$$

so

$$\mathbb{E}[w(E(P_i, \cup_{j>i} P_j)) \leq (1-c/k)^{i-1}B$$

holds for all i. Thus

$$\mathbb{E}[\Delta/2] = \sum_i \mathbb{E}[w(E(P_i, \cup_{j>i} P_j))] \leq \sum_i (1-c/k)^{i-1}B = kB/c. \quad \square$$

Claim 2. In every iteration of the step 2 (`Replacing Expanding Sets with Sets from` $\mathcal{S}$), Δ is decreased by at least $2B$. (Proof by picture!)

Hence step 2 takes polynomial time in expectation, and after its completion, $\mathbb{E}[\Delta] \leq 2kB/c$, so $\mathbb{E}[B'] \leq 2B/c$. Also for all i,

$$\delta(P_i) \leq 2B \leq B' \leq 2B'/\epsilon.$$

When step 3 (`Aggregating`) terminates, we have

$$\begin{aligned} |P_i| &\leq 2(1+\epsilon)n/k \quad \forall i \\ \sum_i |P_i| &= n \\ \delta(P_i) &\leq 2B'/\epsilon \quad \forall i \\ \sum_i \delta(P_i) &\leq kB' \end{aligned}$$

and no two of the P_i's can be merged without violating these constraints. So by The Aggregation Lemma, the number of (nonempty) sets is at most

$$\frac{kB'}{2B'/\epsilon}+\frac{n}{2(1+\epsilon)n/k}+\max\left\{\frac{kB'}{2B'/\epsilon}, \frac{n}{2(1+\epsilon)n/k}, 1\right\} = \frac{\epsilon k}{2}+\frac{k}{2(1+\epsilon)}+\frac{k}{2(1+\epsilon)} \leq k.$$

This completes the proof since in the final solution,

$$\mathbb{E}[\max \delta(P) : P \in \mathcal{P}] \leq 2\mathbb{E}[B']/\epsilon \leq 4B/c\epsilon. \quad \square$$

Proof of Theorem 2. Run `Algorithm 1` to get the family $\mathcal{S}$, and then run `Algorithm 2` (with $B = \max\{\delta(S) : S \in \mathcal{S}\}$) to get the desired partition $\mathcal{P}$. By Lemma 1, the conditions of Lemma 2 are satisfied (with $B = \alpha\,\mathrm{OPT}$ and $c = 1/5\gamma$) and Algorithm 2 outputs a partition with

$$\mathbb{E}[\max \delta(P) : P \in \mathcal{P}] \leq 4B/c\epsilon \leq (20\alpha\gamma/\epsilon)\,\mathrm{OPT} = O\left(\sqrt{\log n \log k}\right)\mathrm{OPT}\,.$$

The algorithm can be generalized to handle additional constraints: a family of terminal sets that must be separated by the partition, and one can also put an upper bound for the sum boundary sizes. See Section 4 of the original paper for the details.

Problem (Min-Max-Multiway-Cut). Given edge-weighted graph G and special vertices $t_1, \ldots, t_k$, partition $V(G)$ into $S_1, \ldots, S_k$ such that $\forall i : \; t_i \in S_i$ so as to minimize $\max \delta(S_i)$.

The best known algorithm for this problem has approximation factor $O(\log^2 n)$ [Svitkina and Tardos'04]. The generalized algorithm results in a (true) $O(\sqrt{\log n \log k})$-approximation algorithm for Min-Max-Multiway-Cut.