"Minimize the Union: Tight Approximations for Small Set Bipartite Vertex Expansion"

Vaggos Chatziafratis SUNet ID: vaggos

June 12, 2017

1 Introduction: What is this paper about?

The problem we will study here [CDM17] is a variation of a really famous and well-studied problem with many applications: the *Maximum-k-Cover* problem. As many of you know, the problem (informally) asks you the following:

- Given a ground set U, a set system $S \subseteq 2^U$ on U, and an integer k, choose exactly k sets from S in order to maximize the number of elements of U that are covered.

For the *Maximum-k-Cover*, the natural greedy algorithm[1] that typically one sees in an undergraduate algorithms class achieves a $(1 - 1/e)$-approximation and this is actually tight [Fei98]. The problem we study here is the minimization version of *Maximum-k-Cover*: it is called *Minimum-k-Union* (Min-k-Union, for short) and despite being a natural problem, surprisingly, it was not studied until very recently [CDK$^+$16]. It asks for the following:

- Given a ground set U, a set system $S \subseteq 2^U$ on U, and an integer k, choose exactly k sets from S in order to minimize the number of elements of U that are covered.

How can we solve this problem? A first natural approach would be to use the analogous greedy algorithm[2], but it turns out that this can be arbitrarily bad, since greedy is completely "blind" to a collection of k sets $\mathcal{C}$ that all have slightly more elements than greedy's current choice, even though if the sets in $\mathcal{C}$ may be identical and would actually constitute the optimal solution to Min-k-Union. This motivates looking at the intersection of the sets $S \subseteq 2^U$ and asking for a collection of subsets with *dense* intersection. Does this problem remind you of anything?

It turns out that this is related to a well-studied problem called the Densest-k-Subgraph [BCC$^+$10] and even though we will mainly focus on lower bounds for the Min-k-Union, we will mention some connections between the two problems that you should keep in mind. Before we proceed, however, we will need to think of Min-k-Union in an equivalent way in terms of a bipartite graph and the associated problem is called *Small Set Bipartite Vertex Expansion* (SSBVE):

- In the *Small Set Bipartite Vertex Expansion* problem (SSBVE) we are given a bipartite graph (U, V, E) with $n = |U|$ and $n' = |V|$ and an integer $k \leq |U|$. The goal is to return a subset $S \subseteq U$ with $|S| = k$ minimizing the expansion[3] $|N(S)|/k$ (equivalently, minimizing $|N(S)|$).

Note that whenever we pick a vertex from U we *have to* count all of its neighbours in our solution. This is an important difference with Densest-k-Subgraph (D-k-S). Let's first recall the definition of D-k-S:

- In the *Densest k Subgraph* problem (D-k-S) the objective is to find the maximum density[4] subgraph on exactly k vertices. This problem generalizes the clique problem and is thus NP-hard in general graphs.

[1] Greedy starts with the $\varnothing$ set and at each step, greedily picks sets that will cover as many elements as possible, given the previously chosen sets.
[2] Here, in each step, we pick sets that have small contribution in terms of new elements added in the final solution.
[3] For a graph $G = (V, E)$ and a subset $S \subseteq V$, the neighborhood $N(S)$ is defined as $N(S) = \{v : \exists (u, v) \in E \text{ with } u \in S\}$.
[4] In terms of number of edges between the chosen k vertices.

Note that here, in contrast to SSBVE, whenever we pick a vertex, we won't have to include all of its neighbours in our solution. This asymmetry in SSBVE between the U and V, in other words, means that for Min-k-Union, we are looking not for an arbitrary dense subgraph of a bipartite graph but rather for a dense subgraph of the form $S \cup N(S)$ (where S is a subset of U of size k), since once we choose the set $S \subseteq U$, we *must* take into account all neighbours of S in V.

Example: Suppose that there are k nodes in U with degree r whose neighbourhoods overlap on some set of $r-1$ nodes of V , but each of the k nodes also has one neighbour that is not shared by any of the others. Then a D-k-S algorithm (or any straightforward modification) might return the bipartite subgraph induced by those k nodes and their $r-1$ common neighbours. But even though the intersection of the k neighbourhoods is large, making the returned subgraph very dense, their union is much larger (since k could be significantly larger than r). So taking those k left nodes as our SSBVE solution would be terrible, as would any straightforward pruning of this set.

This example shows that we cannot simply use a D-k-S algorithm, and there is also no reduction which lets us transform an arbitrary SSBVE instance into a D-k-S instance where we could use such an algorithm. Instead, in order to achieve the upper bound of the paper, the authors fundamentally change previous approaches to tackle densest subgraphs type of problems to take into account the aforementioned asymmetry of SSBVE.

1.1 Today's Results

Today we will mainly focus on lower bounds for the Min-k-Union problem, even though the paper also focuses on upper bounds. We briefly mention the results we will touch on today:

- We will give some convex relaxations for the Min-k-Union problem. We start with a natural Linear Programming relaxation LP and also give the natural semidefinite programming relaxation SDP and the relaxation based on the Sherali-Adams hierarchy.
- We will see that the SDP is bad having an integrality gap of $\tilde{\Omega}(\sqrt{n})$ for the family of random bipartite graphs.
- For the same family of instances (random bipartite graphs), we will see that the SA hierarchy of superconstant rounds is slightly better than the SDP having an integrality gap of $\tilde{\Omega}(n^{1/4})$.
- Apart from the above unconditional lower bounds against certain convex relaxations, we will briefly see a conditional lower bound for Min-k-Union.
- We will briefly see how the authors achieve an $O(n^{1/4+\epsilon})$ (for any constant ϵ) approximation for Min-k-Union.

2 Relaxing the Problem

Now that we have defined the problem and seen some related facts, we can start thinking of how we can solve it. We already observed that the greedy algorithm will not perform well. Perhaps, what we can do is formulate a convex relaxation of the problem. Unfortunately we will show here that this approach, relaxing to an SDP or the Sherali-Adams hierarchy (SA) of slightly superconstant number of rounds will both have large integrality gaps. As an easier first step, before we give the SDP or the SA relaxation, we formulate our problem as a linear program (LP).

2.1 Min-k-Union via LP

The bipartite graph is $G = (U, V, E)$. We have a variable $x_v \in \{0, 1\}$ for every vertex. The intended semantics is that $x_v = 0$ if the vertex v is not part of the k nodes $\in U$ we picked and also is not a neighbour of any of the k nodes $\in U$ we picked; $x_v = 1$ if either we picked x_v or it is neighbour of a picked node. With this in mind the LP relaxation is the following:

$$\text{minimize} \sum_{v \in V} x_v$$

$$\text{subject to:} \sum_{u \in U} x_u \geq k$$

$$x_v \geq x_u, \forall (u,v) \in E, u \in U, v \in V$$

$$0 \leq x_w \leq 1, \forall w \in U \cup V$$

Note that the objective function only depends on the nodes of V that are picked. The first constraint forces that exactly k nodes on the left side U are picked, since it is a minimization problem. The $x_v \geq x_u$ constraint implies that if a node $x_u \in U$ is picked to be part of the Min-k-Union solution, then all its neighbours (i.e. all x_v's for $(u,v) \in E$) also need to be part of the solution.

2.2 Min-k-Union via SDP

We will see two versions of a semidefinite relaxation. The ideas for the objective and the constraints are similar to the ones we already saw for the LP, but as is usual in SDPs, here we will associate a vector $\mathbf{w}$ for every vertex $w \in U \cup V$. To keep notation simple, when a vertex is part of U then the associated vector will be denoted as $\mathbf{u}$ and when a vertex v is part of V the associated vector will be $\mathbf{v}$. The first and easier version of the SDP is the following SDP1:

$$min \sum_{v \in V} ||\mathbf{v}||^2$$

$$s.t. \sum_{u \in U} ||\mathbf{u}||^2 = k$$

$$\mathbf{u} \cdot \mathbf{v} = ||\mathbf{u}||^2, \forall (u,v) \in E, u \in U, v \in V$$

$$\mathbf{w}_1 \cdot \mathbf{w}_2 \geq 0, \forall w_1, w_2 \in U \cup V$$

It is easy to see why the above is a relaxation of the Min-k-Union problem. In the intended integral optimal solution OPT with objective OPT, for all the vertices from U that are picked we can assign to them the standard basis vector $e_1 = (1, 0, ..., 0)$ and also do exactly the same for their neighbours. Then the objective would be OPT as all vertices of OPT would have been assigned a unit vector. The first constraint, that we take exactly k nodes from U and the second about the inner products are also satisfied for the same reasons. The third constraint is satisfied since all inner products are either 0 or 1.

The above relaxation SDP1 has integrality gap $\tilde{\Omega}(\sqrt{n})$ where $n = |U|$, i.e. the number of nodes on the left side U. We will shortly prove this result, however, we want to include the SDP (SDP2) that is actually used in the paper, even though all the ideas and intuition needed for that slightly harder version will already be present in our proofs; we just present the analysis for SDP1 which is easier for the benefit of presentation.

In SDP2, the idea is to have a tighter relaxation with extra constraints concerning the average of the vectors assigned on the nodes of U and for that we introduce one extra vector $\mathbf{v}_0$. Specifically, SDP2 is the following:

$$min \sum_{v \in V} ||\mathbf{v}||^2$$

$$s.t. \sum_{u \in U} ||\mathbf{u}||^2 = k$$

$$\mathbf{u} \cdot \mathbf{v} = ||\mathbf{u}||^2, \forall (u,v) \in E, u \in U, v \in V$$

$$\mathbf{w}_1 \cdot \mathbf{w}_2 \geq 0, \forall w_1, w_2 \in U \cup V$$

$$\sum_{u \in U} \mathbf{u} = k\mathbf{v}_0$$

$$||\mathbf{v}_0||^2 = 1$$

$$\mathbf{w} \cdot \mathbf{v}_0 = ||\mathbf{w}||^2, \forall w \in U \cup V$$

The proof presented in the paper is that the above SDP2, though it is tighter than SDP1, it still has integrality gap $\tilde{\Omega}(\sqrt{n})$ where $n = |U|$.

2.3 Min-k-Union via Sherali-Adams

We now proceed with convex relaxations that extend the linear programming relaxation we saw before. The idea is that we can have a lot of variables that try to capture information not only about the singleton nodes as in the LP (variables we had there, were the x_u, x_v) but instead have variables x_S for sets S of the nodes. The SA hierarchy of r rounds is the following SA:

$$\text{minimize} \sum_{v \in V} x_{\{v\}}$$

$$s.t. \sum_{u \in U} x_{S \cup \{u\},T} \geq k x_{S,T}, \forall S, T \subseteq U \cup V : |S| + |T| \leq r$$

$$x_{S \cup \{v\},T} \geq x_{S \cup \{u\},T}, \forall (u, v) \in E, u \in U, v \in V$$

$$0 \leq x_{S,T} \leq 1, \forall S, T \subseteq U \cup V : |S| + |T| \leq r$$

$$x_\varnothing = 1$$

For one round ($r = 1$) we get back the LP relaxation since then we would still only get constraints for the singleton variables. More generally, we have to define what the variable x_S and $x_{S,T}$ are and also convince ourselves that the above SA is indeed a relaxation of the Min-k-Union problem.

The variables x_S are defined for all subsets $S \subseteq U \cup V$ of size at most r and we also have auxiliary variables $x_{S,T}$ that are just linear combinations of variables x_S:

$$x_{S,T} = \sum_{J \subseteq T} (-1)^{|J|} x_{S \cup J},$$

for subsets S, T such that $|S| + |T| \leq r$. In the intended integral solution, $x_S = \Pi_{w \in S} x_w$ and $x_{S,T} = \Pi_{w \in S} x_w \Pi_{w \in T}(1 - x_w)$. Note that $x_{S,T} = 0$ if $S \cap T \neq \varnothing$ and also $x_{S,\varnothing} = x_S$. Having the above interpretation in mind for the variables that appear in the SA relaxation and the fact that we relax the integrality constraint ($0 \leq x_{S,T} \leq 1$), it should be clear that SA is indeed a valid relaxation of the Min-k-Union problem. It turns out that the integrality gap for $r = O(\epsilon \log n / \log \log n)$ rounds is $\tilde{\Omega}(n^{1/4 - O(\epsilon)})$.

3 Lower Bounds

Now that we saw the candidate relaxations for our problem, the goal of this section is to prove **unconditional lower bounds** but against certain families of algorithms, namely the SDP and SA from above. Specifically, we will prove the following about our relaxations:

Theorem (SDP): The integrality gap for both the SDPs (SDP1 and SDP2) above is $\tilde{\Omega}(\sqrt{n})$.

Theorem (SA): The integrality gap for the SA hierarchy with $r = O(\epsilon \log n / \log \log n)$ rounds is $\tilde{\Omega}(n^{1/4})$.

Note that the SDP appears to be worse than the SA hierarchy of superconstant rounds and also that the lower bound for SA matches the upper bound for Min-k-Union that is proved in the paper.

3.1 Integrality Gap Instances

Here we describe the family of graph inputs where we have a large integrality gap. It turns out that the same family of inputs that cause a large integrality gap for the SDP, will cause a large integrality gap for the SA hierarchy as well. The family of graph inputs is the following:

- Random Bipartite Graphs $G(U, V, E)$: The set of nodes on the left will be denoted by U and the set of nodes on the right by V.
- We pick $U = n$, $V = \sqrt{n}$ and the number of sets to select for the Min-k-Union to be $k = \sqrt{n}$ (although these choices lead to the claimed integrality gap, an analysis similar to the one we do here can be done more generally).
- An edge between two vertices $u \in U$ and $v \in V$ is present with probability $p = \frac{c \log n}{\sqrt{n}}$.

3.2 Optimal solution of the integrality gap instances

Here we state and prove 3 key properties of an integral solution of such a random bipartite graph. The first one is a lower bound for the value of the optimal combinatorial solution.

Lemma 1: The cost of the combinatorial optimal solution OPT is at least $\sqrt{n}/2$ w.h.p.

Proof. Suppose the contrary, that the optimal solution has cost less than $\sqrt{n}/2$. Then, there exists a subset $S \subseteq U$ of size $\sqrt{n}/2$ and a subset $T \subseteq V$ of size $\leq \lfloor\sqrt{n}/2\rfloor$ such that all of the neighbours of S are inside this "small" set T on the right side of the graph, i.e. $N(S) \subseteq T$ (S will be then the optimal solution). Let us now bound the probability that $N(S) \subseteq T$ for **fixed sets** S and T with $|S| = \sqrt{n}/2$ and $|T| = \lfloor\sqrt{n}/2\rfloor$.

If $N(S) \subseteq T$ then there are no edges between S and $V \setminus T$. There are at least $|S| \cdot |T| = n/2$ pairs of vertices and the probability that there is no edge between any of them is at most $(1-p)^{n/2} \leq e^{-pn/2}$. There are at most $n^{\sqrt{n}}$ ways to pick a set S and at most $n^{\sqrt{n}/2}$ to pick a set T and so by the union bound:

$$Pr\left(\exists S, T : N(S) \subseteq T\right) \leq n^{3\sqrt{n}/2} e^{-pn/2}$$

Choosing a large enough constant c in the edge probability $p = \frac{c \log n}{\sqrt{n}}$, we get the lemma. □

The next lemma is used in the analysis of the SDP integrality gap (the tighter version SDP2) and helps us make the graph regular.

Lemma 2: Every vertex in U has degree between $c \log n/2$ and $3c \log n/2$ and every vertex in V has degree between $c\sqrt{n} \log n/2$ and $3c\sqrt{n} \log n/2$.

Proof. Just observe that the average degree for a vertex in U is $c \log n$ and for a vertex in V is $c\sqrt{n} \log n$. The lemma follows by a simple Chernoff bound followed by a union bound across all vertices. □

Finally, the following lemma is used in the analysis of the SA hierarchy integrality gap and it will help us prove an upper bound on the solution produced by the SA relaxation.

Lemma 3: Every two vertices in U are connected by a path of length 4.

Proof. Consider two vertices $u_1, u_2 \in U$. We can assume using Lemma 2, that $deg(u_1) \geq 2, deg(u_2) \geq 2$. Choose a random neighbour $v_1 \in V$of u_1, and a random neighbour $v_2 \in V \setminus \{u_1\}$ of u_2. We are going to show that v_1 and v_2 have a common neighbor $u' \in U \setminus \{u_1, u_2\}$ with high probability (given v_1 and v_2), and, therefore, u_1 and u_2 are connected with a path $u_1 \to v_1 \to u' \to v_2 \to u_2$ of length 4 with high probability. Note that events $\{(v_1, u') \in E\}$ for all $u' \in U \setminus \{u_1, u_2\}$ and events $\{(v_2, u') \in E\}$ for all $u' \in U \setminus \{u_1, u_2\}$ are independent. Therefore, the probability that for a fixed $u' \in U, (v_1, u') \in E$ and $(v_2, u') \in E$ is $p^2 = \frac{c^2 \log^2 n}{n}$ and thus, the probability that for some $u' \in U$ we have $(v_1, u') \in E$ and $(v_2, u') \in E$ is $1 - (1-p^2)^{n-2} \geq 1 - e^{\Omega(\log^2 n)}$. The lemma follows. □

3.3 Lower Bound 1: the SDPs are bad

We start by proving a $\tilde{\Omega}(\sqrt{n})$ lower bound for the SDPs, which means that the natural SDPs are not at all good.

Theorem (SDP): The integrality gap ($k = \sqrt{n}$) for both the SDPs (SDP1 and SDP2) above is $\tilde{\Omega}(\sqrt{n})$.

Proof. We will give the proof for SDP1 and after that we will provide some insights on how to prove the result for the tighter relaxation SDP2 that has more constraints. Our strategy for both proofs has two main ingredients:

- We will try, as is common in the integrality gap proofs, to provide specific values for the variables (the node vectors) of the SDP1 and then check that they satisfy the **constraints** and also that the **objective value** is small. Note that since Lemma 1 tells us that the cost of the combinatorial optimal solution OPT is at least $\sqrt{n}/2$ w.h.p., in order to prove the desired gap, we will need to argue that the SDP1 solution has cost at most $O(poly \log) = \tilde{O}(1)$.
- How can one find a lot of node vectors simultaneously? Recall the fact that a p.s.d. matrix X arises as the Gram matrix of some vectors, i.e. every entry of X is an inner product between two vectors. We will use this idea, and present an appropriate matrix X so that its entries satisfy the constraints and then we will prove that this matrix is indeed p.s.d. Using the above fact the matrix X essentially will provide the candidate solution for the SDP1 and we will be done.

Note that the matrix X needs to have dimension $(n+\sqrt{n}) \times (n+\sqrt{n})$ since there are that many nodes in the graph, each of which needs to be assigned a vector. Guided by the constraints, a natural thing to do is to assign the vectors' mass *uniformly*, in some sense, in order to satisfy them and also have small objective value. More specifically:

- Let $X = \begin{bmatrix} A & C \\ C^T & B \end{bmatrix}$. Let A be $n \times n$, and B be $\sqrt{n} \times \sqrt{n}$. Note that X is symmetric and that the matrix C is $n \times \sqrt{n}$.
- For $u_1, u_2 \in U$, let $\mathbf{u}_1 \cdot \mathbf{u}_2 = a_{u_1 u_2} = 1/\sqrt{n}$ if $u_1 = u_2$. Let's briefly forget for now what values should be assigned to non-diagonal entries and we will return to that shortly. For $v_1, v_2 \in V$, let $\mathbf{v}_1 \cdot \mathbf{v}_2 = b_{v_1 v_2} = \frac{4c^2 \log^2 n}{\sqrt{n}}$ (where c is the constant in the edge probability $p = \frac{c \log n}{\sqrt{n}}$) if $v_1 = v_2$. Again, forget for now the non-diagonal entries and we will return to that shortly. Finally, for $u \in U, v \in V$ let $\mathbf{u} \cdot \mathbf{v} = c_{uv} = 1/\sqrt{n}$ if $(u, v) \in E$. Forget for now entries for the case $(u, v) \notin E$.
- Note that in some sense we try to satisfy every constraint uniformly since we treat every vertex in the same way so far and also have a small objective value; a posteriori we know that the objective value will be $O(\log^2 n)$ and that's why we assigned this "mass" uniformly on the nodes of V (note here that the objective only depends on vectors assigned to vertices in V).

Now we proceed by checking everything that needs to be checked in order for a solution to be feasible:

- Value of SDP: $\sum_{v \in V} ||\mathbf{v}||^2 = \sum_{v \in V} b_{vv} = \frac{4c^2 \log^2 n}{\sqrt{n}} |V| = \tilde{O}(1)$ (recall $|V| = \sqrt{n}$).
- The sum $\sum_{u \in U} ||\mathbf{u}||^2 = \sum_{u \in U} a_{uu} = \sum_{u \in U} 1/\sqrt{n} = \frac{|U|}{\sqrt{n}} = \frac{n}{\sqrt{n}} = \sqrt{n} = k$ as we wanted.
- For every edge $(u, v) \in E$ we have: $\mathbf{u} \cdot \mathbf{v} = c_{uu} = 1/\sqrt{n} = a_{uu} = ||\mathbf{u}||^2$ as we wanted.
- Obviously, $\mathbf{w}_1 \cdot \mathbf{w}_2 \geq 0$ since all entries of the matrices A, B, C (even the entries we didn't yet reveal) are non-negative.

□

This concludes the proof that SDP1 has an integrality gap of $\tilde{\Omega}(\sqrt{n})$, modulo two points: The first one is what are the entries of the matrices that we didn't reveal and the second is why is the matrix X p.s.d. anyways. We now deal with these two related issues and what we will say next actually implies that the SDP2 has the same integrality gap (just by checking the extra constraints) even though it is tighter than SDP1:

- Our goal for choosing the values is to make the matrix X p.s.d. We need to reverse engineer the right values but the short answer here is to just give you the correct values to assign to the remaining entries, which is for $u_1 \neq u_2$: $a_{u_1u_2} = \frac{n/2}{d_L(d_R-1)n} v_{u_1u_2} + \frac{n/2-\sqrt{n}}{n(n-1)}$, where $d_L = c\log n$ is the average left degree, $d_R = \sqrt{n}c\log n$ is the average right degree and $v_{u_1u_2}$ is the *number of common neighbours of vertices* u_1, u_2. For matrix B, we assign $\frac{2c^2\log^2 n}{\sqrt{n}}$ for the off-diagonal entries and for matrix C we assign $(4c^2\log^2 n - d_R/\sqrt{n})/(n-d_R)$ for $(u,v) \in E$.
- The trick to prove that X is p.s.d. is to write it as a sum of 3 p.s.d. matrices: $X = D + Y + W$ where D is a diagonal matrix with nonnegative diagonal entries (and therefore p.s.d.) and Y, W are p.s.d. This rewriting of the matrix X requires some algebraic calculations and manipulations which are not interesting, however, at the core of the proof is the following cute trick in order to prove that a matrix is p.s.d. which we separately prove as a lemma and which is useful to keep in mind.

Useful Lemma: Suppose A is an $n \times n$ matrix which is constant a, D an $m \times m$ matrix which is constant d and B, C^T are constant $n \times m$ matrices (b, c respectibely). Then, $X = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$ is p.s.d. if and only if the 2×2 matrix $M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ is p.s.d.

Proof. The proof is easy just by computing the quadratic forms of the matrices, however this easy lemma is at the heart of the proof of the paper. For example, for $n = 3$, $m = 2$ the matrix becomes:

$$X = \begin{bmatrix} a & a & a & b & b \\ a & a & a & b & b \\ a & a & a & b & b \\ c & c & c & d & d \\ c & c & c & d & d \end{bmatrix}$$

and observe that for a vector $v = [v_1, v_2, v_3, v_4, v_5]$, the two quadratic forms $v^T X v$ and $[v_1 + v_2 + v_3, v_4 + v_5] M [v_1 + v_2 + v_3, v_4 + v_5]^T$ are equal:

$$v^T X v = [v_1, v_2, v_3, v_4, v_5] \begin{bmatrix} a & a & a & b & b \\ a & a & a & b & b \\ a & a & a & b & b \\ c & c & c & d & d \\ c & c & c & d & d \end{bmatrix} [v_1, v_2, v_3, v_4, v_5]^T = [v_1, v_2, v_3, v_4, v_5] \cdot \begin{bmatrix} a(v_1+v_2+v_3) + b(v_4+v_5) \\ a(v_1+v_2+v_3) + b(v_4+v_5) \\ a(v_1+v_2+v_3) + b(v_4+v_5) \\ c(v_1+v_2+v_3) + d(v_4+v_5) \\ c(v_1+v_2+v_3) + d(v_4+v_5) \end{bmatrix} =$$

$$[v_1 + v_2 + v_3, v_4 + v_5] \cdot \begin{bmatrix} a(v_1+v_2+v_3) + b(v_4+v_5) \\ c(v_1+v_2+v_3) + d(v_4+v_5) \end{bmatrix} = [v_1 + v_2 + v_3, v_4 + v_5] M [v_1 + v_2 + v_3, v_4 + v_5]^T.$$

□

3.4 Lower Bound 2: the Sherali-Adams LP

The main result of the corresponding section of the paper is:

Theorem (SA): The integrality gap for the SA hierarchy with $r = O(\epsilon \log n / \log\log n)$ rounds is $\tilde{\Omega}(n^{1/4})$.

Recall that the variables x_S of the SA relaxation are defined for all subsets $S \subseteq U \cup V$ of size at most r and we also have auxiliary variables $x_{S,T}$ that are just linear combinations of variables x_S:

$$x_{S,T} = \sum_{J \subseteq T} (-1)^{|J|} x_{S \cup J},$$

for subsets S, T such that $|S| + |T| \leq r$.

Here we sketch (really high level) the result of the paper about the SA relaxation which is an integrality gap of $\tilde{\Omega}(n^{1/4})$. For this section we let $r \leq \epsilon \log n / \log\log n$, $a = \frac{1}{2r}$, $b = a^r$ and $k = b\sqrt{n}/4 = n^{1/2 - O(\epsilon)}$. For a set $S \subseteq U \cup V$, we denote $S_U = S \cap U$ and $S_V = (S \cap V) \setminus N(S_U)$. The authors introduce the following definition that will be part of the assignment of the variables in the SA relaxation:

Definition: Let us say that $(\mathcal{T}, S')$ is a cover for a set $S \subseteq U \cup V$ if $\mathcal{T}$ is a tree in G (possibly, $\mathcal{T}$ is empty), $S' \subseteq S_V$ (possibly, $S' = \varnothing$), and each vertex in $S_U \cup S_V$ lies in $\mathcal{T}$ or in S'; we require that if $\mathcal{T}$ is not empty, it contains at least one vertex from U. The cost of a cover $(\mathcal{T}, S')$ is $|\mathcal{T} \cap U| + |S'| + 1$. A minimum cover of S is a cover of minimum cost; we denote the cost of a minimum cover by $cost(S)$.

Having this definition in mind we now have all the ingredients to set the variables x_S that appear in the SA relaxation:

$$x_S = b^{|S_U|} a^{|S_V|} \frac{1}{n^{cost(S)/4}}$$

Although in the case of the SDPs, checking that the proposed solution is indeed feasible was relatively easy, just by checking some constraints on the vectors associated with the nodes, here the situation is a bit different. The exact analysis is out of the scope of today's presentation but we will say that through a series of lemmas the authors establish some kind of monotonicity properties between variables $x_{S \cup \{u\}}$ and x_S for "small" sets S (sets of size at most r) and using these monotonicity properties they are able to show feasibility for the SA relaxation and also that the objective value is $O(n^{1/4})$ and hence get the integrality gap of $\tilde{\Omega}(n^{1/4})$.

3.5 Conditional Lower Bound

So far, we saw *unconditional* lower bounds, but against specific algorithms, namely the natural SDP relaxation and the SA hierarchy with superconstant number of rounds; these lower bounds give some indication that the problem at hand is difficult. Here, we highlight a connection between Densest-k-Subgraph and Min-k-Union and give a brief sketch of a $\tilde{\Omega}(n^{1/4})$ lower bound. Note that this lower bound matches the lower bound we got for the SA hierarchy with $O(\log n / \log\log n)$ rounds and as we see in the next section, where we describe the algorithmic result of the paper, it also matches the current best upper bound for Min-k-Union.

For the *conditional* lower bound, guided by the connections that Min-k-Union has with the Densest-k-Subgraph problem, the authors explore the so-called **"log-density"** framework that was developed for Densest-k-Subgraph. This is a general algorithm framework, first introduced in [BCC$^+$10] and we can summarise the main ideas as follows:

- Begin by considering the problem of distinguishing between a random structure (for D-k-S, a random graph) and a random structure in which a small, statistically similar solution has been planted.
- Several algorithmic techniques (both combinatorial and LP/SDP based) fail to solve such distinguishing problems, and thus the gap (in the optimum) between the random and planted case can be seen as a natural lower bound for approximations.
- To match these lower bounds algorithmically, one develops robust algorithms for this distinguishing problem for the case when the planted solution does have statistically significant properties that would not appear in a pure random instance, and then, with additional work, adapts these algorithms to work for worst-case instances while guaranteeing the same approximation.

Along those lines, people have studied the problem of distinguishing between random instances and random instances with some planted structure. We define the log-density of a graph on n nodes to be $\log_n(D_{avg})$, where D_{avg} is the average degree. One of the famous problems is the **DENSE vs RANDOM** problem which is parameterized by k and constants $0 < a, b < 1$: Given a graph G, distinguish between the following two cases: 1) $G = G(n, p)$ where $p = n^{a-1}$ (and thus the graph has log-density concentrated around a), and 2) G is adversarially chosen so that the densest k-subgraph has log-density b where $k^b >> pk$ (and thus the average degree inside this subgraph is approximately k^b. The following conjecture implies that the known algorithms for D-k-S are tight:

Conjecture: For all $0 < a < 1$, for all sufficiently small $\epsilon > 0$, and for all $k \leq \sqrt{n}$, we cannot solve **DENSE vs RANDOM** with log-density a and planted log-density b in polynomial time (w.h.p.) when $b \leq a - \epsilon$.

Using a hypergraph analog of this conjecture (since the Min-k-Union is a hypergraph analog for the smallest m-Edge subgraph problem, which is in turn the minimization version of D-k-S), the authors are able to prove as an easy corollary (just by setting parameters appropriately) the following:

Conditional Lower Bound: For any constant $\epsilon > 0$, there is no polynomial-time algorithm which can distinguish between the two cases from **DENSE vs RANDOM** (the analogous hypergraph conjecture) when the gap between the Min-k-Union objective function in the two instances is $\Omega(m^{1/4-\epsilon})$ (m here denotes the number of edges in the graph). By transforming to the SSBVE problem, we get a similar gap of $\Omega(n^{1/4-\epsilon})$ (n is the number of nodes on the left side). This also clearly implies the same gap for the worst-case setting.

4 Upper bounds - Brief Overview

Complementing the above lower bound, the authors are able to show that in the random planted setting, one can appropriately modify the basic structure of the algorithmic approach of [BCC+10] and achieve an $O(n^{1/4+})$-approximation for any constant $\epsilon > 0$, matching the above lower bound for this model. However, the main technical contribution is an algorithm which matches this guarantee in the worst case setting:

Theorem (Upper Bound): For any constant $\epsilon > 0$ there is a polynomial time $O(n^{1/4+\epsilon})$ approximation algorithm for SSBVE.

We will not prove this here, as it is a complicated algorithm with an involved analysis. It is based however, in the previous framework of [BCC+10]. One of the obstacles for transferring the technology already developed for D-k-S to SSBVE in a graph $G = (U, V, E)$ (or equivalently to Min-k-Union) is the asymmetry we already mentioned in the introduction between the sets that we are allowed to pick in the two problems: In SSBVE once we pick a set S from U, we are forced to take all of its neighbours as part of our solution whereas in D-k-S this is not the case and we already saw an example (see Section 1) where this arises to a serious problem. The example shows that due to this asymmetry, simply using D-k-S is not enough and actually one novel aspect of the authors' approach is a new asymmetric pruning idea which allows to isolate a relatively small set of nodes in V which will be responsible for collecting all of the "bad" neighbours of small sets which would otherwise have small expansion. Even with this tool in place, they still need to trade off a number of procedures in each step to ensure that if the algorithm halts it will return a set that is both small and has small expansion (ignoring the pruned set on the right).

References

[BCC+10] Aditya Bhaskara, Moses Charikar, Eden Chlamtac, Uriel Feige, and Aravindan Vijayaraghavan. Detecting high log-densities: an o (n 1/4) approximation for densest k-subgraph. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 201–210. ACM, 2010.

[CDK+16] Eden Chlamtác, Michael Dinitz, Christian Konrad, Guy Kortsarz, and George Rabanca. The densest k-subhypergraph problem. *arXiv preprint arXiv:1605.04284*, 2016.

[CDM17] Eden Chlamtác, Michael Dinitz, and Yury Makarychev. Minimizing the union: Tight approximations for small set bipartite vertex expansion. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 881–899. SIAM, 2017.

[Fei98] Uriel Feige. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.