Max Cut for random graphs with a planted partition

B. Bollobás [*†] A.D. Scott [‡]

Abstract

We give an algorithm that, with high probability, recovers a planted k-partition in a random graph, where edges within vertex classes occur with probability p and edges between vertex classes occur with probability $r \geq p + c\sqrt{p \log n/n}$. The algorithm can handle vertex classes of different sizes and, for fixed k, runs in linear time. We also give variants of the algorithm for partitioning matrices and hypergraphs.

1 Introduction

Graph problems such as Max Cut, Max k-Cut and Min Bisection are well-known to be NP-hard (see Garey, Johnson and Stockmeyer [17] and Garey and Johnson [16]); indeed even approximating Max Cut or Max k-Cut to within an factor $(1 + o(1))$ is NP-hard, although the approximation complexity of Min Bisection remains open (see Papadimitriou and Yannakakis [25], Håstad [18] and Kann, Khanna, Lagergren and Panconesi [23]). For random graphs $G \in \mathcal{G}(n, p)$, on the other hand, it is often easy to find an approximate solution quickly and with high probability: for instance, the value of Max Cut is, with high probability, $np^2/4 + O(n^{3/2} \log n)$, while the simple greedy algorithm that, one vertex at a time, puts each vertex into a class maximizing the number of cross-edges will find a cut of size at least $e(G)/2$, which is close to optimal provided p is not too small. (For very sparse random graphs, the results are a little different: see Coppersmith,

[*]Trinity College, Cambridge CB2 1TQ *and* Department of Mathematical Sciences, University of Memphis, Memphis TN38152; email: bollobas@msci.memphis.edu

[†]Research supported in part by NSF grant ITR 0225610 and DARPA grant F33615-01-C-1900

[‡]Department of Mathematics, University College London, Gower Street, London WC1E 6BT; email: scott@math.ucl.ac.uk

Gamarnik, Hajiaghayi and Sorkin [11]. For random dense graphs, there is a polynomial time approximation scheme for Max Cut: see Arora, Karger and Karpinski [3] and Frieze and Kannan [15].)

It is therefore interesting to consider random graphs with parameters such that there are likely to be cuts that are significantly larger than the expected value size of a random cut (for graphs with the same density). Two models of random graphs have frequently been considered in this context. A random graph G in the $\mathcal{G}(n, m, b)$ model is chosen uniformly at random from all graphs with n vertices, m edges and maximum cut of size b. A second model, which we shall consider in this paper, involves choosing a partition in advance, and then adding edges so that, with high probability, the chosen partition will be a maximum cut. More precisely, if $\{V_i\}_{i=1}^k$ is a partition of $V(G)$, a random graph G with *planted k-partition* $V(G) = \bigcup_{i=1}^k V_i$ *and parameters* (p, r) is obtained by taking edges within each class V_i independently with probability p, and edges between classes independently with probability r. Thus we expect the planted partition to be a good cut of G, provided $r - p$ is not too small.

In this paper, we shall consider Max Cut and Max k-Cut for random graphs with planted partitions, where p and r are chosen so that the planted partition is almost surely the unique optimal cut; in particular, we will have $p < r$. Much of the previous literature is concerned with Min Bisection for graphs with a planted bipartition, and of course $p > r$. With minor modifications, our results and proofs transfer easily to that context.

Random graphs with small bisections were considered by Bui, Chaudhuri, Leighton and Sipser [8], who showed that, for random d-regular graphs with minimum bisection size b, an optimal bisection can be found in polynomial time with high probability, provided $b = o(n^{1-1/\lfloor (d+1)/2 \rfloor})$. Dyer and Frieze [12] gave an algorithm that finds an optimal bisection of a random graph G with m edges in expected polynomial time, provided $m = \Omega(n^2)$ and the optimal bisection has size at most $(1 - \epsilon)m/2$. Boppana [6] gave a polynomial-time algorithm that finds an optimal bisection with probability $1 - O(1/n)$, provided $m = \Omega(n \log n)$ and G has a bisection of size at most $m/2 - 5\sqrt{mn \log n}/2$.

We now turn to the planted bisection model with parameters (p, r) (see Condon and Karp [10] for additional discussion). Jerrum and Sorkin investigated the performance of the Metropolis algorithm on this model in [20], proving that if $p - r \geq n^{-1/6+\epsilon}$ then the algorithm finds an optimal bisection in time $O(n^3)$, with probability $1 - \exp(-n^{\Omega(\epsilon)})$. Juels [21] analyzed a simple hill-climbing algorithm, and showed that if $p - r = \Omega(1)$, then the algorithm find the planted partition with high probability in time $O(n^2)$.

A number of authors have worked on finding algorithms that allow $p-r$ to be as small as possible. Kučera [24] showed that if $p-r \geq c\sqrt{\log n/n}$ then it is possible to partition a small set of vertices in time $O(n \log n)$ and then use this partial partition to find the planted partition with high probability. Carson and Impagliazzo [9] gave an $O(n^2)$ algorithm that finds an optimal bisection with high probability provided $p - r = \omega((p \log^2 n/n)^{1/2})$. Boppana [6] also gives a (high-degree) polynomial time algorithm that succeeds almost surely provided $p - r \geq c\sqrt{p \log n/n}$. Finally, Feige and Kilian [13] also gave a polynomial-time algorithm that finds an optimal bisection with high probability provided $p - r \geq c\sqrt{p \log n/n}$; in addition, their algorithm is robust against an adversary who can add edges inside vertex classes and delete edges between vertex classes. Note that these algorithms are essentially optimal up to the constant: if $p - r = o(\sqrt{p \log n/n})$ then the planted bisection will not in general be the optimal bisection.

Planted partitions with more than two classes were considered by Condon and Karp [10], who showed that, if $p - r = \Omega(n^{-1/2+\epsilon})$, a planted partition with a fixed number of classes of equal size can be recovered in linear time and with probability $1 - \exp(-n^{\Theta(\epsilon)})$. Ben-Dor, Shamir and Yakhini [5] showed that, with k classes of size $\Omega(n)$, it is possible to find the planted partition in time $O(n^2/\log^c n)$, provided $p - r = \Omega(1)$. Shamir and Tsur [27] gave an $O((k/\log n + 1)n^2)$ time algorithm that with high probability finds a planted partition with $k = O(\sqrt{n}/\log n)$ classes of equal size, provided $p - r \geq k \log n/\sqrt{n}$, and finds a planted partition where each vertex class has at least $\sqrt{n \log n}/(p-r)^{1+\epsilon}$ vertices in time $O(kn^2/\log n)$, provided $p - r \geq O(n^{-1/2+\epsilon})$.

In this paper we address a variety of planted partition problems in which different classes may have different sizes. We give an algorithm that runs in time $O(km + n)$, and recovers a planted partition in a graph of order n with high probability provided $p-r \geq c\sqrt{p \log n/n}$, where c depends on the minimum density $|V_i|/n$ of vertex classes V_i in the partition. Our algorithm has several advantages over previous algorithms: it is fast (running in time $O(km+n)$) and comparatively simple, and does not require the size of vertex classes to be specified in advance. The algorithm is also easily adapted to related problems such as partitioning hypergraphs or Boolean matrices with planted partitions. We remark that we do not know whether the algorithm works against an adversary who can add edges between vertex classes and delete edges within classes. We conjecture that our algorithm is robust against such an adversary.

The algorithm is, as far as we know, novel in several ways. Previous algorithms have used a variety of techniques, including random walk meth-

ods and using a partition of a subgraph to partition the whole graph. Our algorithm breaks the graph up into a large number ($O(\sqrt{\log n})$) of separate pieces, and partitions them in what we hope is a successively more accurate manner, in each case using the previous partition to guide our partition of the next piece. In order to maximize our use of information, the algorithm revisits and repartitions each piece on a number of occasions: we use disjoint collections of edges one each occasion, so these separate visits give 'independent' partitions. The algorithm is also 'self-starting', in that we do not need a good partition to get it off the ground, but instead just begin with a random partition.

Our algorithm, and its variants, are given in section 2. In section 3 we state some useful lemmas, and then in sections 4 and 5 we prove the correctness of the main algorithm. For simplicity, we ignore floors and ceilings throughout, as this has no significant affect on our analysis. It seems likely that the algorithms will work successfully (with high probability) for a much larger range of parameters than our analysis covers. Thus the algorithm may well provide an effective practical heuristic for finding large cuts. As most of the algorithms are variants or applications of the core algorithms Algorithm A and B, we give detailed proofs for these two algorithms and state the remaining results without proof.

Finally, we note briefly that another way of looking at the problem is that we have a partition of G that we wish to *hide*. We must have $|p - r|$ sufficiently large so that the partition is almost surely the unique cut (or k-cut) of maximum size; however, we also want to choose $|p - r|$ small enough to make the partition hard to detect. Problems of this type when the concealed structure is a large clique have been considered from an algorithmic perspective by Feige and Krauthgamer [14] and Alon, Krivelevich and Sudakov [1], and from a cryptographic viewpoint by Juels and Peinado [22]; a similar problem for Hamiltonian cycles was considered by Broder, Frieze and Shamir [7]. The results below suggest that the pair (p, r) must be chosen with great care for the hidden partition to be both verifiable and secure, ie the cut must be large enough to be unlikely in a random graph but small enough that it does not 'stand out'. Indeed, it is not clear that such a pair exists.

2 The algorithm, and its variants

The basic idea of the algorithm is as follows. Suppose for simplicity that we have a planted partition $V(G) = V_1 \cup V_2$, where V_1 and V_2 have equal

size, with internal edges having probability p and cross-edges having probability $r > p$. We choose an integer M and divide $V(G)$ randomly into sets $S_1, \dots, S_M$. Let $R_1 \cup B_1$ be a random partition of S_1 into two sets of equal size. The difference $|R_1 \cap V_1| - |B_1 \cap V_1|$ has standard deviation $\Theta(\sqrt{n/M})$, so we are likely to have an imbalance of size at least $\Omega(\sqrt{n/M})$ between $|R_1 \cap V_1|$ and $|B_1 \cap V_1|$. Suppose that this occurs and, say, $|R_1 \cap V_1| > |B_1 \cap V_1|$. Now since $p < r$, it follows that, for $v \in V_1 \cap S_2$, we have $\mathbb{E}|\Gamma(v) \cap B_1| > \mathbb{E}|\Gamma(v) \cap R_1|$, while the vertices in $V_2 \cap S_2$ satisfy the opposite inequality. We therefore attempt to use our partition $S_1 = R_1 \cup B_1$ to generate a more imbalanced partition $R_2 \cup B_2$ of S_2: let R_2 be the vertices in S_2 with more neighbours in B_1 than R_1, and let B_2 be the vertices in S_2 with more neighbours in R_1. Provided the colour imbalance between R_1 and B_1 is large enough (and $|p - r|$ is not too small), we should expect this to create a larger colour imbalance between R_2 and B_2. Thus (after deleting some vertices to make sure that $|R_2| = |B_2|$) we obtain sets R_2, B_2 of equal size and with a greater imbalance than $R_1 \cup B_1$. We now use $R_2 \cup B_2$ to produce a partition $R_3 \cup B_3$ of S_3, and so on. At each stage, the imbalance between colours is amplified until, by the time we reach S_M, we expect to be getting the correct partition at each stage. Finally, we use the partition of S_M to partition $V(G) \setminus S_M$, and combine these last two partitions to give a partition of $V(G)$.

Needless to say, there are several obstacles to making this work as it stands: for small $\Delta = r - p$, we have to allow M to grow with $n = |G|$, and so the average size of the S_i must be $o(n)$. In order to keep the S_i as large as possible, it is helpful to be able to 'reuse' the sets: since the edges between S_i and S_j are independent from the edges between other pairs of sets, we can treat a given set of vertices as 'independent' on each visit. Thus, for instance, we could use S_1 to partition S_2, then S_2 to partition S_3, and then S_3 to give a new partition of S_1. We maximize the gain from this by reusing sets as many times as possible. In addition, although sets S_i have size $o(n)$, we use two sets of size $\Omega(n)$ at the end: this reduces the probability of misplacing vertices in the final stages of the algorithm (our final partition must have no errors, while we are tolerant of a few errors at earlier stages).

We are now ready to state the first algorithm (for Max Cut). Note that, in order avoid clutter, we omit floors and ceilings throughout the paper (thus, for instance, we write $n/40\sqrt{\log n}$ instead of $2\lfloor n/80\sqrt{\log n}\rfloor$), and tacitly assume that n is sufficiently large for our bounds to make sense. This does not have any significant effect on the analysis.

Algorithm A (Split(α))**.**

Input: *A graph G with n vertices and a parameter α.*
Output: *A bipartition of $V(G)$.*

Step 0. Partition $V(G)$ at random into $M = 20\sqrt{\log n}$ sets $T_1, \dots, T_M$ of size $n/40\sqrt{\log n}$ and two sets T_{M+1}, T_{M+2} of size $n/4$.

Let $R_1 \cup B_1$ be a random partition of T_1 into two sets of equal size.

Step 1. Let $v_1, \dots, v_L$ be an Euler circuit of the complete graph K_M with vertex set $[M]$ (or K_M without a matching if M is even), and define $S_i = T_{v_i}$ for $i = 1, \dots, L$. Let $S_{L+1} = T_{M+1}$, $S_{L+2} = T_{M+2}$ and $S_{L+3} = V(G) \setminus S_{L+2}$.

Step 2. For $i = 1, \dots, L+2$, we partition S_{i+1} into R_{i+1} and B_{i+1} as follows.

If $\min\{|R_i|, |B_i|\} < \alpha|S_i|/6$ then the algorithm halts and reports failure. Otherwise, we remove randomly chosen vertices from the larger class until we obtain sets $Q_1^{(i)} \subset R_i$ and $Q_2^{(i)} \subset B_i$ with $|Q_1^{(i)}| = |Q_2^{(i)}| = \min\{|R_i|, |B_i|\}$.

We place $v \in S_{i+1}$ into B_{i+1} if $|\Gamma(v) \cap Q_1^{(i)}| > |\Gamma(v) \cap Q_2^{(i)}|$, into R_{i+1} if $|\Gamma(v) \cap Q_2^{(i)}| > |\Gamma(v) \cap Q_1^{(i)}|$, and otherwise assign v with equal probability to B_{i+1} or R_{i+1}.

Step 3. Output the partition $(R_{L+2} \cup R_{L+3}, B_{L+2} \cup B_{L+3})$.

Note that in Steps 1 and 2, each set T_j occurs repeatedly among the S_i and is therefore visited many times; however, T_j is partitioned independently at each visit.

When applied to a random graph with a planted k-partition (and suitable parameters), Algorithm A produces (with high probability) a bipartition in which each vertex class of the output bipartition entirely contains at least one colour class from the planted partition. Thus two colour classes from the planted partition have been correctly assigned (one to each side of the output bipartition), although other colour classes may have vertices on both sides of the output bipartition.

We shall usually assume that we have a random graph with n vertices and planted partition $V(G) = \bigcup_{i=1}^{k} V_i$, where $|V_i| \geq cn$ for every i and the planted partition has parameters (p, r). The parameters $p = p(n)$, $r = r(n)$,

$\Delta = \Delta(n)$ and $c = c(n)$ will satisfy the following inequalities.

$$p(1-p) \geq 8000 \log n / c^2 n \tag{1}$$

$$r = p + \Delta \geq p + \frac{40000}{c^3}\sqrt{\frac{p \log n}{n}} \tag{2}$$

$$c \geq 200(\log n)^{-1/8}. \tag{3}$$

The constants in (1)-(3) are far from optimal for the results below; indeed, they could be substantially improved at the cost of more detail in the proofs.

Theorem 1. *Let G be a random graph with planted partition satisfying* (1)-(3). *Then in time $O(e(G)+n)$ and with probability $1 - O(n^{-5})$ Algorithm A with $\alpha = c$ finds a partition $V(G) = W_1 \cup W_2$ such that W_1 and W_2 each contain at least one vertex class V_j from the planted partition.*

Here, and in the sequel, a larger value of Δ will reduce the failure probability: in particular, increasing the constants increases the power of $1/n$ in the bound on failure probability.

A minor modification of Algorithm A allows us to deal with cases when $r < p$: we simply exchange B_{i+1} and R_{i+1} at the end of each iteration of Step 2. The analysis is then identical to the proof of Theorem 1, except that the roles of r and p are interchanged. We must therefore replace (1) and (2) by

$$r(1-r) \geq 8000 \log n / c^2 n \tag{4}$$

$$p = r + \Delta \geq r + \frac{40000}{c^3}\sqrt{\frac{r \log n}{n}}. \tag{5}$$

Similar statements hold for the other results in the paper.

When there are only two colour classes, Algorithm A enables us to recover a planted bipartition, and therefore solve Max Cut with high probability for random graphs with a planted partition and suitable parameters.

Corollary 2. *Let G be a random graph with planted bipartition satisfying* (1)-(3). *Then in time $O(e(G)+n)$ and with probability $1-O(n^{-5})$ Algorithm A with $\alpha = c$ recovers the planted bipartition.*

In order to deal with more than two colours, we must modify Algorithm A to obtain a bipartition in which *every* vertex class in the planted partition appears only on one side of the final bipartition. The modified algorithm is the same as Algorithm A, except that we keep track of how many vertices

are pushed strongly towards one side of the partition at each stage: when we come to partition S_{i+1} in step 2 of Algorithm A, a vertex vertex $v \in V_j \cap S_{i+1}$ is likely to have a strong preference towards one side of the partition if our partition of $V_j \cap S_i$ is very biased. Thus we can recognize (with high probability) which vertices belong to a colour that is very biased. At an appropriate stage, we push all vertices that do not have a strong preference (and consequently most vertices from the corresponding vertex classes) onto one side of the partition, which creates a strong preference in every colour class at subsequent stages.

One difficulty here is to pick the appropriate stage. We want to push vertices across at a stage when every colour class is either strongly biased, or is fairly evenly split: vertices in classes of the first type are in general not moved (and are mostly on one side in any case), while vertices in classes of the second type are mostly pushed onto one side. We must therefore pick a stage at which there are no colour classes with 'moderate bias'. We can do this by looking one stage into the future: with high probability, colour classes that are strongly biased at stage i will remain strongly biased at stage $i+1$, while colour classes that are moderately biased at stage i will become strongly biased at stage $i+1$. Thus if there is a moderately biased colour class at stage i, the proportion of vertices seeing a large imbalance at stage $i+1$ among their neighbours in S_i will increase significantly. The algorithm checks for this by looking ahead, and then steps a stage back to push vertices across.

Algorithm B (Perfectsplit(α))**.**

Input: *A graph G with n vertices and a parameter α.*
Output: *A bipartition of $V(G)$.*

Step 0. Set $I = \lfloor L - \log n \rfloor$. Run Algorithm A, proceeding as far as $i = I$ in Step 2.

Step 1. For each $v \in S_I$, calculate

$$D_v := |\Gamma(v) \cap Q_1^{(I)}| - |\Gamma(v) \cap Q_2^{(I)}|.$$

Order the vertices in decreasing order of $|D_v|$, and let D^* be the value of $|D_v|$ for the $(\alpha|S_I|/2)$-th vertex. Let

$$\Delta^* = \frac{D^*}{|Q_1^{(I)}|}$$

be the difference in densities from v to $Q_i^{(I)}$ and $Q_2^{(I)}$.

Step 2. We now continue running Step 2 of Algorithm A, for $i > I$, with an additional calculation for each i: we say that a vertex $v \in S_{i+1}$ has large imbalance at stage i *if*

$$D_v \geq \frac{\Delta^* |Q_1^{(i)}|}{(\log n)^{1/20}}.$$

We let L_{i+1} be the set of vertices in S_{i+1} with large imbalance after stage i, set $H_i = S_{i+1} \setminus L_{i+1}$, and define

$$l_{i+1} = \frac{|L_{i+1}|}{|S_{i+1}|}.$$

Step 3. Let i^* be the first value of $i > I$ with $l_{i+1} < l_{i-1} + \alpha/2$ (if no such i is found, the algorithm halts and reports failure). We backtrack one step and replace R_{i^*} and B_{i^*} by

$$R'_{i^*} = R_{i^*} \cap L_{i^*}$$

and

$$B'_{i^*} = B_{i^*} \cup H_{i^*} = S_{i^*} \setminus R_{i^*}.$$

Step 4. We then run the remainder of Algorithm A, starting with the bipartition $R'_{i^*} \cup B'_{i^*}$ in place of $R_{i^*} \cup B_{i^*}$.

We have the following result.

Theorem 3. *Let G be a random graph with planted partition satisfying* (1)-(3). *Then in time $O(e(G) + n)$ and with probability $1 - O(n^{-5})$ Algorithm B with $\alpha = c$ finds a nontrivial partition of $V(G)$ into two unions of vertex classes from the planted partition.*

Applying Theorem 3 recursively gives the following algorithm for Max k-Cut.

Algorithm C (Partition(α))**.**

Input: *A graph G with n vertices and a parameter α.*
Output: *A partition of $V(G)$.*

Step 0. Run Algorithm B to obtain a bipartition $W_1 \cup W_2$.

Step 1. Recursively run Algorithm B on each W_i, omitting step 1 (and keeping the same value of Δ^* from the first iteration). If there is no l_i with $l_i > \alpha/6$ then output W_i as a colour class; otherwise, W_i is partitioned as $W_i = X_1 \cup X_2$, and we repeat this step on each X_i.

This algorithm determines the original planted partition with high probability.

Theorem 4. *Let G be a random graph with planted partition satisfying* (1)-(3). *Then in time $O(e(G)k + n)$ and with probability $1 - O(n^{-4})$ Algorithm C with $\alpha = c$ finds the planted partition.*

Proof. Note that successive iterations of the algorithm cannot quite be treated as independent: the partition of $V_1 \cup V_2$ determines which collections of colour classes the algorithm is applied to in the next iteration. However, it is sufficient if the algorithm works on all 2^k subsets of the colour classes, which is true with probability at least $1 - O(n^{-4})$ since $2^k = o(n)$. □

A related problem was given by Condon and Karp [10], who raised the question of recovering planted Boolean matrix partitions. Let $R_1 \cup R_2$ be a partition of $[m]$ and $C_1 \cup C_2$ be a partition of $[n]$. A *planted Boolean matrix with partition $(R_1, R_2; C_1, C_2)$ and parameters (p, r)* is a random $m \times n$ matrix with 0-1 entries chosen independently, where $\mathbb{P}(M[i,j] = 1) = p$ if $(i,j) \in (R_1 \times C_1) \cup (R_2 \times C_2)$ and $\mathbb{P}(M[i,j] = 1) = r$ otherwise. A small modification of Algorithm A works in this context.

Algorithm D (Matrixsplit(α))**.**

Input: *A Boolean matrix with m rows and n columns, and a parameter α.*
Output: *A bipartition of $[m]$ and a bipartition of $[n]$.*

Step 0. Let $M = 20\sqrt{\log n}$, and let $[m] = \bigcup_{i=1}^{M+1} T_i$ and $[n] = \bigcup_{i=1}^{M+1} U_i$ be two random partitions such that $|T_1| = m/40\sqrt{\log n}$ for $i = 1, \ldots, M$, $U_i = n/40\sqrt{\log n}$ for $i = 1, \ldots, M$, $|T_{M+1}| = m/2$ and $|U_{M+1}| = n/2$.

Let $R_1 \cup B_1$ be a random partition of T_1 into two sets of equal size.

Step 1. Let $v_1, \ldots, v_L$ be an Euler circuit of $K_{M,M}$ (we may assume that M is even), and define $S_i = T_{v_i}$ for i odd and $S_i = U_{v_i}$ for i even. Let $S_{L+1} = U_{M+1}$, $S_{L+2} = [m]$ and $S_{L+3} = [n]$.

Step 2. Now run Step 1 of Algorithm A as before.

Step 3. Output the partition $(R_{L+2}, B_{L+2}; R_{L+3}, B_{L+3})$.

Providing $r - p$ is sufficiently large, Algorithm D recovers the planted matrix partition with high probability.

Theorem 5. *Let M be a random $m \times n$ Boolean matrix with planted partition $(R_1, R_2; C_1, C_2)$ such that $\min\{|R_1|, |R_2|, |C_1|, |C_2|\} \geq c \max\{m, n\}$. Suppose that $m \geq n$ and that (1)-(3) hold. Let t be the number of non-zero entries in M. Then in time $O(m + n + t)$ and with probability $1 - O(n^{-5})$ Algorithm D with $\alpha = c$ recovers the planted partition.*

The theorem follows by trivial modifications of the proof of Theorem 1. A more general problem, where there are more than two classes, can be dealt with in the same way as for graphs, by modifying the algorithm analogously to Algorithms B and C.

For hypergraphs, a minor modification to Algorithms A and B allows us to produce results similar to Theorems 1 and 3. There are several variants of the problem, but for simplicity let us consider the following: we have a random l-uniform hypergraph H with planted partition $V_1 \cup V_2$, where an edge e is present with probability p if all its vertices belong to the same class V_i, and with probability $r > p$ otherwise. For $x \in V(H)$, and a set $S \subset V(H) \setminus x$, we define the degree $d(x, S)$ of x into S by

$$d(x, S) = |\{e \in E(H) : x \in e, e \subset \{x\} \cup S\}|.$$

We modify Algorithm A as follows.

Algorithm E (Hypersplit(c))**.**

Input: *An l-uniform hypergraph G with n vertices and a parameter α.*
Output: *A bipartition of $V(G)$.*

Step 0. Run steps 0 and 1 as in Algorithm A.

Step 1. We run step 2 as before, except that we place $v \in S_{i+1}$ into B_{i+1} if $d(v, Q_1^{(i)}) > d(v, Q_2^{(i)})$, into R_{i+1} if $d(v, Q_1^i) < d(v, Q_2^i)$, and otherwise assign v with equal probability to B_{i+1} or R_{i+1}.

Step 2. Output the partition $(R_{L+2} \cup R_{L+3}, B_{L+2} \cup B_{L+3})$ as before.

We have the following result.

Theorem 6. *Let H be a random l-uniform hypergraph with planted partition $V_1 \cup V_2$. Suppose that $|G| = n$ and $\min_{i=1,2} |V_i| \geq cn$, where c satisfies (3). Then there is a constant $K = K(l)$ such that if*

$$r = p + \Delta \geq p + \frac{K}{c^3}\sqrt{\frac{p \log n}{n^{l-1}}},$$

and $p(1-p) \geq K \log n / c^2 n$, then in time $O(e(G) + n)$ and with failure probability $O(n^{-5})$ Algorithm E with $\alpha = c$ finds the planted partition.

3 Lemmas

We gather here a few basic inequalities that we shall use in the next section.

We use the following version of Chernoff's inequality (see [19]). Let X be the sum of an independent set of 0-1 Bernoulli random variables with parameters $p_1, \ldots, p_n$, and let $\mu = \sum_{i=1}^n p_i$. Then

$$\mathbb{P}(X \leq \mu - t) \leq e^{-t^2/2\mu} \tag{6}$$

and

$$\mathbb{P}(X \geq \mu + t) \leq \exp\left(-\frac{t^2}{2(\mu + t/3)}\right). \tag{7}$$

We shall also use (see, for instance, [2]) the fact that

$$\mathbb{P}(X \geq \mathbb{E}X + t) \leq \exp(-2t^2/n) \tag{8}$$

and

$$\mathbb{P}(X \leq \mathbb{E}X - t) \leq \exp(-2t^2/n). \tag{9}$$

We will also need the Berry-Esseen inequality (see Petrov [26], pages 111, 128).

Lemma 7. *Let $X_1, \ldots, X_n$ be independent random variables that satisfy $\mathbb{E}X_i = 0$ and $\mathbb{E}|X_i|^3 < \infty$ for $i = 1, \ldots, n$. Let $B = \sum_{i=1}^n \mathbb{E}X_i^2$ and $L = B^{-3/2} \sum_{i=1}^n \mathbb{E}|X_i|^3$. Then, for $x \in \mathbb{R}$,*

$$\mathbb{P}\left(\sum_{i=1}^n X_i < x\sqrt{B}\right) = \Phi(x) + O(L). \tag{10}$$

Furthermore, the $O(L)$ term has absolute value at most $0.8L$.

We shall apply this to the special case when $X_1, \ldots, X_n$ are independent Bernoulli random variables with parameters $p_1, \ldots, p_n$. Taking $Y_i = X_i - p_i$, we have $\mathbb{E}Y_i = 0$, $\mathbb{E}Y_i^2 = p_i(1-p_i)$ and

$$\mathbb{E}|Y_i^3| = (1-p_i)p_i^3 + p_i(1-p_i)^3 = p_i(1-p_i)(p_i^2 + (1-p_i)^2) \leq p_i(1-p_i),$$

so in this case, $\sum_{i=1}^n \mathbb{E}|Y_i|^3 \leq \sum_{i=1}^n \mathbb{E}Y_i^2$. We therefore have $B = \sigma^2 = \sum_{i=1}^n \mathbb{E}Y_i^2 = \sum_{i=1}^n p_i(1-p_i)$ and $L = B^{-3/2}\sum_{i=1}^n \mathbb{E}|Y_i|^3 \leq B^{-3/2} \cdot B = 1/\sigma$, and so Lemma 7 implies the following.

Lemma 8. *Let $X_1, \ldots, X_n$ be independent Bernoulli random variables with parameters $p_1, \ldots, p_n$ and let $\sigma^2 = \sum_{i=1}^n p_i(1-p_i)$. Then, for $x \in \mathbb{R}$,*

$$\left| \mathbb{P}\left(\sum_{i=1}^n X_i < \sum_{i=1}^n p_i + x\sigma \right) - \Phi(x) \right| \leq \frac{0.8}{\sigma}. \tag{11}$$

Lemma 8 will usually suffice to control the distributions in which we are interested. However, we shall sometimes need to deal with very small deviations, when the $O(1/\sigma)$ error term is larger than the deviation we wish to estimate. In this situation, we shall need Corollary 10 below.

First we state a standard fact.

Lemma 9. *The sum of two bounded, centrally symmetric, integer-valued unimodal random variables is centrally symmetric and unimodal.*

Proof. Let $f, g : \mathbb{Z} \to \mathbb{R}_{\geq 0}$ be the densities (with respect to counting measure) of two centrally symmetric and unimodal random variables X and Y. If f and g are characteristic functions of intervals $[-a, a]$ and $[-b, b]$ then the density of $X+Y$ is the convolution $f \star g$, which is clearly centrally symmetric and unimodal. More generally, we can write $f = \sum \lambda_i \chi_i$ and $g = \sum \lambda'_j \chi'_j$ as positive linear combinations of indicator functions of centrally symmetric intervals (in $\mathbb{Z}$): then $f \star g = \sum_{i,j} \lambda_i \lambda'_j \chi_i \star \chi'_j$ is a positive linear combination of centrally symmetric unimodal functions and is therefore centrally symmetric and unimodal. □

Corollary 10. *Suppose $p_1, \ldots, p_n$ are reals in $[0, 1]$, and*

$$X = \sum_{i=1}^n (B(p_i) - B'(p_i)),$$

where $B(p_i)$ and $B'(p_i)$, $1 \leq i \leq n$, are $2n$ independent Bernoulli random variables. Let $\sigma^2 = 2\sum_{i=1}^n p_i(1-p_i)$. Then, for $0 \leq \lambda \leq 4\sigma$,

$$\mathbb{P}(X \leq \lambda) \geq \frac{1}{2} + \frac{1}{2}\mathbb{P}(X = 0) + \frac{\lfloor \lambda \rfloor}{9\sigma} - \frac{\lfloor \lambda \rfloor}{2\sigma^2}.$$

Proof. For $p \in [0,1]$, $B(p) - B'(p)$ is centrally symmetric and unimodal, since $p^2 + (1-p)^2 \geq 2p(1-p)$; by Lemma 9, X is also centrally symmetric and is unimodal (and $\mathbb{E}X = 0$). Since $X + n$ has distribution $\sum_{i=1}^{n}(B(p_i) + B(1-p_i))$ and variance σ^2, it follows from (11) that

$$\begin{aligned}
\mathbb{P}(0 < X \leq 4\sigma) &= \mathbb{P}\left(\mathbb{E}(X+n) < X + n \leq \mathbb{E}(X+n) + 4\sigma\right) \\
&- \mathbb{P}(X + n \leq \mathbb{E}(X+n) + 4\sigma) - \mathbb{P}(X + n \leq \mathbb{E}(X+n)) \\
&\geq \left(\Phi(4) - \frac{0.8}{\sigma}\right) - \left(\Phi(0) - \frac{0.8}{\sigma}\right) \\
&> \frac{4}{9} - \frac{1.6}{\sigma}. \qquad (12)
\end{aligned}$$

Since X is symmetric and unimodal with mean 0, we deduce that, for $0 \leq \lambda \leq 4\sigma$,

$$\begin{aligned}
\mathbb{P}(X \leq \lambda) &\geq \mathbb{P}(X \leq 0) + \mathbb{P}(0 < X \leq \lambda) \\
&\geq \left(\frac{1}{2} + \frac{1}{2}\mathbb{P}(X = 0)\right) + \frac{\lfloor \lambda \rfloor}{4\sigma}\mathbb{P}(0 < X \leq 4\sigma),
\end{aligned}$$

and the result now follows from (12) □

4 Proof of Theorem 1

Our aim in this section is to prove Theorem 1.

We break up the analysis of Algorithm A into a number of lemmas and claims. We remark that the constants in the statements of our claims are by no means best possible.

Let us first state a little notation.

For disjoint sets $X, Y \subset V(G)$, and $v \notin X \cup Y$, let

$$\lambda(v, X, Y) = |\Gamma(v) \cap X| - |\Gamma(v) \cap Y|.$$

We shall often refer to vertex classes V_j in the planted partition as *colour classes*. We define the *bias in colour* j of (X, Y) by

$$\mu_j(X, Y) = |V_j \cap X| - |V_j \cap Y|,$$

and the *imbalance at stage* i by

$$M_i = \max_j |\mu_j(Q_1^{(i)}, Q_2^{(i)})|.$$

Our argument proceeds by examining the sequence $M_1, M_2, \ldots$ of imbalances, and showing that these eventually become large.

Note that since $|Q_1^{(i)}| = |Q_2^{(i)}|$, we have $\sum_{j=1}^{k} \mu_j(Q_1^{(i)}, Q_2^{(i)}) = 0$. So if $\mu_j(Q_1^{(i)}, Q_2^{(i)}) > 0$ then there is a colour j' such that we have $\mu_{j'}(Q_2^{(i)}, Q_1^{(i)}) \geq c\mu_j(Q_1^{(i)}, Q_2^{(i)})$. It follows that there are colours j, j' with

$$\min\{\mu_j(Q_1^{(i)}, Q_2^{(i)}), \mu_{j'}(Q_2^{(i)}, Q_1^{(i)})\} \geq cM_i. \tag{13}$$

Before stating our main sequence of claims, we give the following simple fact. We omit the proof.

Proposition 11. *With probability* $1 - O(\exp(-cn/400\sqrt{\log n}))$ *we have, for every* i *and* j*,*

$$|S_i \cap V_j| \geq c\frac{|S_i|}{2}. \tag{14}$$

Our proof uses a sequence of four claims, which we now state. We then prove that the theorem follows from these claims, before returning to prove the claims.

For the remainder of the proof, we shall assume that (14) holds, so that we are conditioning on this event. More precisely, we condition on the specific values of $S_i \cap V_j$ for all i and j: thus for any collection of values $|S_i \cap V_j|$ satsfying (14), we show that the claims below hold.

The first claim states that all sets R_i and B_i have a reasonable number of vertices (and therefore the algorithm does not fail in step 2); since $|Q_1^{(i)}| = |Q_2^{(i)}| = \min\{|R_i|, |B_i|\}$, the same therefore holds for $Q_1^{(i)}$ and $Q_2^{(i)}$. In addition, $Q_1^{(i)} \cup Q_2^{(i)}$ contains a reasonable number of vertices from each V_j.

More specifically, we will show that, with high probability, for every h we have

$$\min\{|R_h|, |B_h|\} > c|S_h|/6 \tag{15}$$

and

$$\min_j |(Q_1^{(h)} \cup Q_2^{(h)}) \cap V_j| > c^2|S_h|/14. \tag{16}$$

Claim 1. *Suppose that* (15) *is satisfied for* $h = i$*. Then with probability* $1 - O(\exp(-c^4 n/40000\sqrt{\log n}))$ (15) *and* (16) *are satisfied for* $h = i + 1$*.*

The next few claims concern the partition of S_{i+1}, which occurs in the iteration of Step 2 after we have partitioned S_i.

We note first that, for each i, M_i is with constant probability $\Omega(\sqrt{c|S_i|})$: the algorithm needs an imbalance of this size to get off the ground.

Claim 2. *Suppose that* (15) *holds for* $h = i$. *Then*

$$\mathbb{P}(M_{i+1} > \sqrt{c|S_{i+1}|}/10) \geq 1/9.$$

If M_i is not too small, then we hope that the imbalance increases at the next stage, so that $M_{i+1} \gg M_i$. The next claim bounds the probability that this fails to happen.

Claim 3. *If* $\mu_j(Q_1^{(i)}, Q_2^{(i)}) = \mu > 0$ *then the probability that*

$$\mu_j(R_{i+1}, B_{i+1}) < 3 \min\left((\log n)^{1/4}\mu, \frac{|V_j \cap S_{i+1}|}{4}\right)$$

is at most

$$\max\{e^{-2(\log n)^{1/2}\mu^2/|S_i|}, \exp(-cn/12800\sqrt{\log n})\}.$$

Finally, if M_i is large then M_{i+1} is large (with high probability); while if M_i is sufficiently large then (with high probability) all vertices of some colour are forced on to the same side of the partition.

Claim 4. *If* $\mu_j(Q_1^{(i)}, Q_2^{(i)}) \geq c^2|S_i|/32$ *then, for* $v \in V_j \cap S_{j+1}$, *we have*

$$\mathbb{P}(v \notin R_{i+1}) = O(\exp(-\sqrt{\log n}/2)), \tag{17}$$

and

$$\mathbb{P}\left(\mu_j(Q_1^{(i+1)}, Q_2^{(i+1)}) \geq c^2|S_{i+1}|/32\right) \geq 1 - O(e^{-c^2 n/10240\sqrt{\log n}}).$$

If, in addition, $|S_i| \geq n/4$ *then, for* $v \in V_j \cap S_{i+1}$, *we have*

$$\mathbb{P}(v \notin R_{i+1}) = O(n^{-6}),$$

and

$$\mathbb{P}(V_j \cap S_{i+1} \subset R_{i+1}) = 1 - O(n^{-5}).$$

Similar statements to Claim 2, Claim 3 and 4 hold with the two sides of the partition exchanged.

Provided the claims above are correct we finish the proof as follows.

Proof of Theorem 1. Proposition 11 implies that (14) has failure probability at most $O(\exp(-cn/400\sqrt{\log n}))$. We therefore assume that (14) holds throughout the proof.

Note that if, for some j and j', we have

$$\mu_j(Q_1^{(i)}, Q_2^{(i)}) \geq c^2|S_i|/32 \tag{18}$$

and

$$\mu_{j'}(Q_2^{(i)}, Q_1^{(i)}) \geq c^2|S_i|/32 \tag{19}$$

then the same holds for all $i' > i$, by Claim 4, with failure probability at most $O(\exp(-c^2 n/10240\sqrt{\log n}))$. Furthermore, the second part of Claim 4 then implies that V_j is entirely contained in one side of the final partition and $V_{j'}$ is entirely contained in the other side, with failure probability at most $O(n^{-5})$. Thus if (18) and (19) both hold for some i, the algorithm finishes correctly with probability at least $1 - O(n^{-5})$. It is therefore sufficient to show that (18) and (19) are true for some $i \leq 190 \log n$.

We now run through the algorithm stage by stage. We say that $\mathcal{P}_i = (Q_i^{(i)}, Q_2^{(i)})$ is *successful* if one of the following two conditions is satisfied.

- $\mathcal{P}_i$ satisfies the inequality in Claim 2 for some colour j and either $i = 1$ or $\mathcal{P}_{i-1}$ was unsuccessful.
- $\mathcal{P}_{i-1}$ was successful and

$$M_i \geq \min\{c(\log n)^{1/4} M_{i-1}/4, c|V_j \cap S_{i+1}|/8\}.$$

Suppose that the algorithm is successful at $\log n$ consecutive stages. It follows that at the end of the run, we have $M_i \geq c|V_j \cap S_{i+1}|/32$, and therefore that there are colours j, j' with

$$\min\{\mu_j(Q_i^{(i)}, Q_2^{(i)}), \mu_{j'}(Q_2^{(i)}, Q_1^{(i)}\} \geq c^2|V_j \cap S_{i+1}|/32.$$

(We use here the fact that if some colour has imbalance μ in one direction, then some other colour has imbalance at least $c\mu$ in the other direction.) It follows from Claim 4 and (14) that with failure probability $O(\exp(-c^2 n/6400\sqrt{\log n}))$

$$\min\{\mu_j(R_{i+1}, B_{i+1}), \mu_{j'}(B_{i+1}, R_{i+1})\} \geq c|S_i|/8,$$

and so $M_{i+1} \geq c|S_i|/8$, since we obtain $Q_i^{(i+1)}$ and $Q_2^{(i+1)}$ by deleting vertices from one of R_{i+1} and B_{i+1}; thus (18) and (19) hold at the next stage .

It therefore suffices to show that, with failure probability at most $O(n^{-5})$, there are $\log n$ consecutive successful stages among the first $190 \log n$.

We shall divide the stages into *runs* as follows. The first run begins at $i = 1$, and each subsequent run begins at the stage after the previous run

ends. A run terminates with the first unsuccessful stage (in which case it is an *unsuccessful run*), or else after $\log n$ successful stages (in which case it is a *successful run*).

Now at the first stage of a run (with $i = 1$ or where $i-1$ was unsuccessful), we are successful with probability at least $1/9$. If we have been successful at $t+1$ consecutive stages, then

$$\begin{aligned} M_i &\geq \min\left\{\frac{\sqrt{c|S_i|}}{10}\left(\frac{c(\log n)^{1/4}}{4}\right)^t, \frac{c|V_j \cap S_{i+1}|}{8}\right\} \\ &\geq \min\left\{20^t(\log n)^{(t-1)/8}\sqrt{|S_i|}, \frac{c|V_j \cap S_{i+1}|}{8}\right\}. \end{aligned}$$

Now there are colours j, j' such that $\min\{\mu_j(Q_1^{(i)}, Q_2^{(i)}), \mu_{j'}(Q_2^{(i)}, Q_1^{(i)})\} \geq cM_i$. Since $M_{i+1} \geq \min\{\mu_j(R_{i+1}, B_{i+1}), \mu_{j'}(B_{i+1}, R_{i+1})\}$, it follows from Claims 3 and 4 that we are successful at the $(i+1)$-st stage with failure probability at most

$$\begin{aligned} &\max\{\exp(-(\log n)^{1/4}20^{2t}(\log n)^{t/4}c^2), \exp(-cn/12800\sqrt{\log n})\} \\ &\qquad \leq \max\{\exp(-400^t(\log n)^{t/4}), n^{-6}\}. \end{aligned}$$

Now with probability $O(n^{-5})$, we are never unsuccessful after we first reach $t = 5$, so it is sufficient to show that, with failure probability $O(n^{-5})$, some run reaches this point.

Now if we make it past the first stage, it follows from the argument above that the probability of failure in one of the next four stages is at most $4\exp(-400(\log n)^{1/4})$: the probability that this happens on $\log n$ consecutive occasions is at most $O(n^{-5})$. Since each such attempt uses at most five steps of the algorithm, these unsuccessful runs use at most $5\log n$ steps. On the other hand, it follows from (6) that the probability we are successful on the first stage on fewer than $\log n$ out of our first $90\log n$ runs is $O(n^{-5})$. Furthermore, the unsuccessful runs use at most $89\log n$ steps of the algorithm. Thus with failure probability at most $O(n^{-5})$ we reach a point in the first $94\log n$ steps where we are successful at the fifth stage, and we carry on to have a successful run with failure probability at most $O(n^{-5})$. □

The remainder of this section is devoted to proving the claims above. Note first that

$$\Delta \geq \frac{40000}{c^3}\sqrt{\frac{p\log n}{n}} \geq \frac{3200000\log n}{c^4 n}. \tag{20}$$

Note also that, for every i, $|S_i| \geq n/40\sqrt{\log n}$.

All that remains is to prove the claims.

Proof of Claim 1. We begin with the first inequality. For $i = 1$ this is immediate. Now suppose $i > 1$ and $\min\{|R_i|, |B_i|\} > c|S_i|/6$. We show that

$$\mathbb{P}\left(\min\{|R_{i+1}|, |B_{i+1}|\} < c|S_{i+1}|/6\right) < \exp(-cn/3000\sqrt{\log n}).$$

Since $|Q_1^{(i)}| = |Q_2^{(i)}|$ there is a colour j such that $|V_j \cap Q_1^{(i)}| \geq |V_j \cap Q_2^{(i)}|$ for some j. Then for $v \in S_{i+1} \cap V_j$, we have $\mathbb{P}(v \in R_{i+1}) \geq 1/2$ in Step 2; since (14) holds, we have $|V_j \cap S_{i+1}| \geq c|S_{i+1}|/2$, and so $\mathbb{E}|V_j \cap R_{i+1}| \geq c|S_{i+1}|/4$. Then Chernoff's inequality (6) applied to any $c|S_{i+1}|/2$ vertices of $V_j \cap S_{i+1}$ implies that

$$\begin{aligned}
\mathbb{P}(|V_j \cap R_{i+1}| \leq c|S_i|/6) &\leq \exp\left(2\left(c|S_i|/24\right)^2/(c|S_i|/2)\right) \\
&= \exp(-c|S_i|/144) \\
&= O(\exp(-cn/6000\sqrt{\log n})).
\end{aligned}$$

Since the same inequality holds for every R_i and B_i, this implies the first inequality in the claim.

For the second inequality, note that we obtain $Q_1^{(i+1)}$ and $Q_2^{(i+1)}$ by deleting vertices from at most one of R_{i+1} and B_{i+1}. Thus provided (15) holds, every vertex is kept with probability at least $c/6$. It follows from (14) that

$$\mathbb{E}(|(Q_1^{(i+1)} \cup Q_2^{(i+1)}) \cap V_j|) \geq c^2|S_i|/12,$$

and so by Chernoff's inequality we get (16) with failure probability at most

$$\exp\left(-2(c^2|S_i|/84)^2/|S_i|\right) \leq \exp\left(-c^4|S_i|/8000\right) \leq \exp\left(-c^4 n/40000\sqrt{\log n}\right).$$

□

Proof of Claim 2. For $i = 0$ this follows easily from Lemma 8. For $i \geq 1$, note that since $|Q_1^{(i)}| = |Q_2^{(i)}|$ there must be distinct j, j' such that $|V_j \cap Q_1^{(i)}| \geq |V_j \cap Q_2^{(i)}|$ and $|V_{j'} \cap Q_1^{(i)}| \leq |V_{j'} \cap Q_2^{(i)}|$. Then for $v \in S_{i+1} \cap V_j$ and $w \in S_{i+1} \cap V_j$, we have $\mathbb{P}(v \in R_{i+1}) \geq 1/2$ and $\mathbb{P}(w \in B_{i+1}) \geq 1/2$ in Step 2. It follows from (14) and Lemma 7 that with probability at least $1/9$ we have $\mu_j(R_{i+1}, B_{i+1}) \geq \sqrt{c|S_{i+1}|}/10$ and $\mu_{j'}(R_{i+1}, B_{i+1}) \leq -\sqrt{c|S_{i+1}|}/10$. Deleting vertices from (one of) R_{i+1} and B_{i+1}, we decrease at most one of the two imbalances. Thus we must have either $\mu_j(Q_1^{(i+1)}, Q_2^{(i+1)}) \geq \sqrt{c|S_{i+1}|}/10$ or $\mu_{j'}(Q_1^{(i+1)}, Q_2^{(i+1)}) \leq -\sqrt{c|S_{i+1}|}/10$, which implies $M_{i+1} \geq \sqrt{c|S_{i+1}|}/10$. □

Before proving Claim 3, let us analyze how strongly vertices are pushed towards one or the other side of the partition at the ith step.

For $i \geq 1$ and $v \in S_{i+1} \cap V_j$, and $h = 1, 2$, we have

$$\mathbb{E}|\Gamma(v) \cap Q_h^{(i)}| = p|V_j \cap Q_h^{(i)}| + (p+\Delta)|Q_h^{(i)} \setminus V_j| = p|Q_h^{(i)}| + \Delta|Q_h^{(i)} \setminus V_j|$$

and so, since $|Q_1^{(i)}| = |Q_2^{(i)}|$,

$$\begin{aligned} \mathbb{E}\lambda(v, Q_1^{(i)}, Q_2^{(i)}) &= \Delta(|Q_1^{(i)} \setminus V_j| - |Q_2^{(i)} \setminus V_j|) \\ &= \Delta(|V_j \cap Q_2^{(i)}| - |V_j \cap Q_1^{(i)}|) \\ &= -\Delta\mu_j(Q_1^{(i)}, Q_2^{(i)}). \end{aligned} \tag{21}$$

Now, since $Q_1^{(i)} \cup Q_2^{(i)} \subset S_i$, we have

$$\begin{aligned} \mathrm{var}\lambda(v, Q_1^{(i)}, Q_2^{(i)}) &\leq p(1-p)|S_i \cap V_j| + (p+\Delta)(1-p-\Delta)|S_i \setminus V_j| \\ &\leq (p+\Delta)|S_i| \end{aligned} \tag{22}$$

while, by (1) and (16),

$$\begin{aligned} \mathrm{var}\lambda(v, Q_1^{(i)}, Q_2^{(i)}) &\geq p(1-p)|(Q_1^{(i)} \cup Q_2^{(i)}) \cap V_j| \\ &\geq \frac{8000 \log n}{c^2 n} \frac{c^2 n}{5600\sqrt{\log n}} \\ &\geq \sqrt{\log n}. \end{aligned} \tag{23}$$

Thus writing σ for the standard deviation of $\lambda(v, Q_1^{(i)}, Q_2^{(i)})$, we have $\sigma \leq \sqrt{(p+\Delta)|S_i|}$ and $\sigma \to \infty$ as $n \to \infty$ in (11).

It follows from (21) and (11) that, for $v \in S_{i+1} \cap V_j$, we have

$$\mathbb{P}(v \in R_{i+1}) \geq \Phi\left(\frac{\Delta\mu_j(Q_1^{(i)}, Q_2^{(i)})}{\sigma}\right) - \frac{1}{\sigma} \tag{24}$$

Proof of Claim 3. Let $\mu = \mu_j(Q_1^{(i)}, Q_2^{(i)})$. It follows from (21) that for $v \in V_j \cap S_{i+1}$ we have

$$\mathbb{E}\lambda(v, Q_1^{(i)}, Q_2^{(i)}) = -\Delta\mu.$$

We split the proof of Claim 3 into three cases, depending on the size of $\Delta\mu$. The first two cases deal with the situation when $\Delta\mu$ is smaller than 2σ. If $\Delta\mu$ is not too small then (24) gives a sufficiently good bound for us to control the distribution of $\mu_j(R_{i+1}, B_{i+1}$. If $\Delta\mu$ is very small, however, the

$O(1/\sigma)$ error term may overwhelm our estimate, so we use a slightly more careful estimate of the distribution of $\lambda(v, Q_1^{(i)}, Q_2^{(i)})$ near its mean. Finally, if $\Delta\mu$ is more than 2σ, we simply use Chebyshev's inequality.

Note that $\Phi(t) \geq 1/2 + t/18$ for $0 \leq t \leq 4$; it follows from (23) that $\sigma > 9$, provided n is sufficiently large.

Case 1: $18 \leq \Delta\mu \leq 2\sigma$

If $v \in S_{i+1} \cap V_j$ then it follows from (24) that

$$\begin{aligned}\mathbb{P}(v \in R_{i+1}) &\geq \Phi\left(\frac{\Delta\mu}{\sigma}\right) - \frac{1}{\sigma} \\ &\geq \frac{1}{2} + \frac{\Delta\mu}{9\sigma} - \frac{1}{\sigma} \\ &\geq \frac{1}{2} + \frac{\Delta\mu}{18\sigma}.\end{aligned}$$

Thus

$$\mathbb{E}(|V_j \cap R_{i+1}|) \geq |V_j \cap S_{i+1}| \left(\frac{1}{2} + \frac{\Delta\mu}{18\sigma}\right). \tag{25}$$

Now by (22) and (14), since $|S_{i+1}| \geq |S_i|$ for every i,

$$\begin{aligned}\frac{\Delta\mu}{18\sigma}|V_j \cap S_{i+1}| &\geq \frac{\Delta\mu c|S_i|/2}{18\sqrt{(p+\Delta)|S_{i+1}|}} \\ &\geq \frac{c\Delta\mu}{36}\sqrt{\frac{|S_{i+1}|}{p+\Delta}} \\ &\geq \frac{c\Delta\mu}{36}\min\left\{\sqrt{\frac{|S_{i+1}|}{2p}}, \sqrt{\frac{|S_{i+1}|}{2\Delta}}\right\} \\ &= \frac{c\mu}{36}\sqrt{\frac{|S_{i+1}|}{2}}\min\left\{\frac{\Delta}{\sqrt{p}}, \sqrt{\Delta}\right\} \\ &\geq \frac{c\mu}{36}\sqrt{\frac{|S_{i+1}|}{2}}\min\left\{\frac{40000}{c^3}\sqrt{\frac{\log n}{n}}, \frac{1789}{c^2}\sqrt{\frac{8000\log n}{n}}\right\} \\ &\geq 5(\log n)^{1/4}\mu. \end{aligned} \tag{26}$$

It follows from (25) that

$$\mathbb{E}(|R_{i+1} \cap V_j| - |B_{i+1} \cap V_j|) \geq 5(\log n)^{1/4}\mu,$$

and so, by (9), the probability that $|R_{i+1} \cap V_j| - |B_{i+1} \cap V_j| \leq 3(\log n)^{1/4}\mu$ is at most

$$\mathbb{P}(|R_{i+1} \cap V_j| \leq \mathbb{E}|R_{i+1} \cap V_j| - (\log n)^{1/4}\mu) \leq e^{-2(\log n)^{1/2}\mu^2/|S_i|}. \quad (27)$$

Case 2: $\Delta\mu \leq 18$

For $v \in S_{i+1} \cap V_j$, we analyze the random variable $\lambda(v, Q_1^{(i)}, Q_2^{(i)})$ as follows. We begin by picking as many pairs of vertices of the same 'type' as possible from $Q_1^{(i)}$ and $Q_2^{(i)}$: let U be a maximal subset of $Q_1^{(i)} \cup Q_2^{(i)}$ such that

$$|U \cap Q_1^{(i)} \cap V_j| = |U \cap Q_2^{(i)} \cap V_j| \quad (28)$$

and

$$|U \cap Q_1^{(i)} \cap V \setminus V_j| = |U \cap Q_2^{(i)} \cap V \setminus V_j|. \quad (29)$$

Thus U contains the same number of vertices from V_j and $V \setminus V_j$ in each class $Q_h^{(i)}$, and hence

$$\lambda(v, U \cap Q_1^{(i)}, U \cap Q_2^{(i)})$$

is symmetric. Furthermore, since $\mu > 0$, the remaining vertices in $P_i := Q_1^{(i)} \setminus U$ are all from V_j and the remaining vertices in $Q_i := Q_2^{(i)} \setminus U$ are all from $V \setminus V_j$. Then $|P_i| = |Q_i| = \mu$, and

$$\lambda(v, Q_1^{(i)}, Q_2^{(i)}) = \lambda(v, U \cap Q_1^{(i)}, U \cap Q_2^{(i)}) + \lambda(v, P_i, Q_i).$$

We decompose $\lambda(v, P_i, Q_i)$ as $\sum_{i=1}^{\mu} \lambda(v, x_i, y_i)$, where $P_i = \{x_i, \ldots, x_\mu\}$, $Q_i = \{y_i, \ldots, y_\mu\}$. Note that

$$\mathbb{P}(vx_i \text{ is an edge}) = p$$

and

$$\mathbb{P}(vy_i \text{ is an edge}) = p + \Delta.$$

We generate the random variables $\lambda(v, x_i, y_i)$ in two steps. Let $Z_1, \ldots, Z_\mu$ be random variables with

$$\mathbb{P}(Z_i = 1) = \mathbb{P}(Z_i = -1) = p(1 - \Delta - p)/(1 - \Delta) < p(1 - p) \leq 1/4 \quad (30)$$

and $Z_i = 0$ otherwise, and let $Y_1, \ldots, Y_\mu$ be Bernoulli random variables with

$$\mathbb{P}(Y_i = 1) = 1 - \mathbb{P}(Y_i = 0) = \Delta \quad (31)$$

We set $\lambda(v, x_i, y_i) = -1$ if $Y_i = 1$ and otherwise set $\lambda(v, x_i, y_i) = Z_i$. Thus

$$\lambda(v, x_i, y_i) = -Y_i + (1 - Y_i)Z_i. \tag{32}$$

Note that

$$\begin{aligned}\mathbb{P}(\lambda(v, x_i, y_i) = -1) &= \mathbb{P}(Y_i = 1) + \mathbb{P}(Y_i = 0)\mathbb{P}(Z_i = 1) \\ &= \Delta + p(1 - \Delta - p) \\ &= (1 - p)(\Delta + p)\end{aligned}$$

and

$$\begin{aligned}\mathbb{P}(\lambda(v, x_i, y_i) = 1) &= \mathbb{P}(Y_i = 0)\mathbb{P}(Z_i = -1) \\ &= p(1 - \Delta - p).\end{aligned}$$

Thus $\lambda(v, x_i, y_i)$ has the correct distribution. Furthermore, since $\mathbb{P}(Z_i = 1) = \mathbb{P}(Z_i = -1) \leq 1/4$, Z_i can be written as the difference of two Bernoulli random variables as in Corollary 10; in particular, Z_i is unimodal.

Now let $X_1 = \{i : Y_i = 1\}$ and $X_0 = \{i : Y_i = 0\}$, so

$$\lambda(v, Q_1^{(i)}, Q_2^{(i)}) = \lambda(v, Q_1^{(i)} \cap U, Q_2^{(i)} \cap U) + \sum_{i \in X_0} \lambda(v, x_i, y_i) - |X_1|.$$

Note that $\mathbb{E}|X_1| = \Delta\mu$ and, by Lemma 9,

$$Z = \lambda(v, Q_1^{(i)} \cap U, Q_2^{(i)} \cap U) + \sum_{i \in X_0} \lambda(v, x_i, y_i) \tag{33}$$

is a symmetric, unimodal random variable for any choice of X_0.

Let us condition on the value of $|X_1|$. Provided $|X_1| < 2\sigma$, Corollary 10 implies that, since $|X_1|$ is an integer,

$$\begin{aligned}\mathbb{P}(v \in R_{i+1}) &= \mathbb{P}\left(\mu(v, Q_1^{(i)}, Q_2^{(i)}) < 0\right) + \frac{1}{2}\mathbb{P}\left(\mu(v, Q_1^{(i)}, Q_2^{(i)}) = 0\right) \\ &= \mathbb{P}(Z < X_1) + \frac{1}{2}\mathbb{P}(Z = X_1) \\ &= \mathbb{P}(Z \leq X_1) - \frac{1}{2}\mathbb{P}(Z = X_1) \\ &\geq \frac{1}{2} + \frac{|X_1|}{9\sigma} - \frac{|X_1|}{2\sigma^2} \\ &\geq \frac{1}{2} + \frac{|X_1|}{12\sigma},\end{aligned}$$

since $\sigma \geq 18$. Here we have used the fact that $\mathbb{P}(Z = X_1) \leq \mathbb{P}(Z = 0)$, which follows from the symmetry and unimodality of Z.

Let $X = \min\{|X_1|, 4\sigma\}$. Then, since $\mathbb{P}(v \in R_{i+1})$ clearly increases as $|X_1|$ increases, we have

$$\mathbb{P}(v \in R_{i+1}) \geq \frac{1}{2} + \frac{|X|}{12\sigma}$$

and so

$$\mathbb{E}(|R_{i+1} \cap V_j| - |B_{i+1} \cap V_j|) \geq \frac{|S_{i+1} \cap V_j|}{6\sigma}\mathbb{E}X. \tag{34}$$

We estimate $\mathbb{E}X$ as follows. Clearly, $\mathbb{P}(X_i = i) = \binom{\mu}{i}\Delta^i(1-\Delta)^{\mu-i}$. Let $s_r = r\binom{\mu}{r}\Delta^r(1-\Delta)^{\mu-r}$, so $\mathbb{E}X_1 = \sum_{r=1}^{\mu} s_r$ and $\mathbb{E}X \geq \sum_{r \leq 4\sigma} s_r$. For $r \geq 3\Delta\mu$ we have

$$\frac{s_{r+1}}{s_r} = \frac{r+1}{r}\frac{\mu - r}{r+1}\frac{\Delta}{1-\Delta} = \frac{\Delta(\mu - r)}{(1-\Delta)r} \leq \frac{\Delta\mu}{r} \leq \frac{1}{3}.$$

It follows that, as $4\sigma \geq 54 \geq 3\Delta\mu$,

$$\Delta\mu = \mathbb{E}|X_1| = \sum_{r=1}^{n} s_r \leq \frac{3}{2}\sum_{r \leq 4\sigma} s_r \leq \frac{3}{2}\mathbb{E}X.$$

Therefore, by (34) and the same calculations as in (26),

$$\begin{aligned}\mathbb{E}(|R_{i+1} \cap V_j| - |B_{i+1} \cap V_j|) &\geq \frac{\Delta\mu|S_{i+1} \cap V_j|}{9\sigma} \\ &\geq 5(\log n)^{1/4}\mu.\end{aligned}$$

As in Case 1, Chernoff's inequality implies that (27) holds.

Case 3: $\Delta\mu > 2\sigma$

If $\Delta\mu > 2\sigma$ then we use Chebyshev: for $v \in S_{i+1} \cap V_j$ we have

$$\mathbb{P}(v \in R_{i+1}) \geq 3/4$$

Thus

$$\mathbb{E}(|R_{i+1} \cap V_j| - |B_{i+1} \cap V_j|) \geq \frac{1}{2}|V_j \cap S_{i+1}|.$$

So by (8) and (14) we have $|R_{i+1} \cap V_j| - |B_{i+1} \cap V_j| > |V_j \cap S_{i+1}|/4$, and so $|R_{i+1} \cap V_j| \leq \mathbb{E}|R_{i+1} \cap V_j| - |V_j \cap S_{i+1}|/8$, with failure probability at most

$$\begin{aligned}\exp\left(-2\left(|V_j \cap S_{i+1}|/8\right)^2/|V_j \cap S_{i+1}|\right) &\leq \exp(-c|S_i|/32) \\ &\leq \exp(-cn/12800\sqrt{\log n}).\end{aligned}$$

□

Proof of Claim 4. Let $\mu = \mu_j(Q_1^{(i)}, Q_2^{(i)})$. For $v \in V_j \cap S_{i+1}$, we decompose $\lambda(v, Q_1^{(i)}, Q_2^{(i)})$ as in (30)-(33). We get

$$\lambda(v, Q_1^{(i)}, Q_2^{(i)}) = Y + Z,$$

where Y is the sum of μ independent 0-1 Bernoulli random variables with parameter Δ and Z is the sum of at most $|S_i|/2$ symmetric random variables Z_i with values in $\{-1, 0, 1\}$ and probability at most $2(p+\Delta)$ of being nonzero (note that the number of Z_i depends on Y, but the Z_i are then independent).

Now $\mathbb{E}Y \geq \Delta\mu$, so (6) and (20) imply that

$$\begin{aligned} \mathbb{P}\left(Y \leq \Delta\mu/2\right) &\leq \mathbb{P}(Y \leq \mathbb{E}Y - \Delta\mu/2) \\ &\leq \exp(-\Delta\mu/8) \\ &\leq \exp\left(-c^2\Delta|S_i|/256\right) \\ &\leq \exp(-2000|S_i|\log n/c^2 n). \end{aligned} \tag{35}$$

Let N be the number of Z_i taking nonzero values (conditioning on the value for Y). Then $\mathbb{E}N \leq 2(p+\Delta)|S_i|$, and (7) implies that

$$\begin{aligned} \mathbb{P}(N \geq 4(p+\Delta)|S_i|) &\leq \exp\left(-(2(p+\Delta)|S_i|)^2/\frac{16}{3}(p+\Delta)|S_i|\right) \\ &= \exp(-3(p+\Delta)|S_i|/8) \\ &\leq \exp(-2000|S_i|\log n/c^4 n). \end{aligned} \tag{36}$$

If $N \leq 4(p+\Delta)|S_i|$ then, since Z is the sum of N independent ± 1 Bernoulli random variables with parameter $1/2$, Chernoff's inequality (9) implies that

$$\mathbb{P}\left(Z \leq -\Delta\mu/2\right) \leq \exp(-2(\Delta\mu/4)^2/2N) \leq \exp(-\Delta^2\mu^2/64(p+\Delta)|S_i|).$$

Now, by (20),

$$\frac{\Delta^2}{p+\Delta} \geq \frac{\Delta^2}{\max\{2p, 2\Delta\}} \geq \min\left\{\frac{\Delta^2}{2p}, \frac{\Delta}{2}\right\} \geq \frac{1600000\log n}{c^4 n}, \tag{37}$$

and so

$$\begin{aligned} \mathbb{P}\left(Z \leq -\frac{\Delta\mu}{2}\right) &\leq \exp\left(-\frac{25000\log n}{c^4 n}\frac{\mu^2}{|S_i|}\right) \\ &\leq \exp\left(-24|S_i|\log n/n\right) \end{aligned} \tag{38}$$

The bounds (36), (38) and (35) together imply the first inequality in the theorem, since $|S_i| \geq n/40\sqrt{\log n}$ for every i, and the third inequality when $|S_i| \geq n/4$.

For the second inequality, note that the first inequality and (7) imply that

$$\begin{aligned}\mathbb{P}(|V_j \cap B_{i+1}| > c|V_j \cap S_{i+1}|/16) &\leq \exp(-(c|V_j \cap S_{i+1}|/32)/4)\\ &\leq \exp(-c|S_{i+1}|/256)\end{aligned}$$

On the other hand, (15), (35) and (6) imply that

$$\begin{aligned}\mathbb{P}\left(|V_j \cap Q_1^{(i+1)}| < c|V_j \cap S_{i+1}|/8\right) &\leq \exp(-c|V_j \cap B_i|/40)\\ &\leq \exp(-c^2|S_{i+1}|/80).\end{aligned}$$

It then follows from (14) that

$$\mathbb{P}(\mu_j(Q_i^{i+1}, Q_2^{i+1}) < c^2|S_{i+1}|/32) = O(\exp(-c^2 n/10240\sqrt{\log n}).$$

Finally, the fourth inequality follows directly from the third. □

5 Proof of Theorem 3

In this section we complete the proof of Theorem 3. We divide the proof into four claims. In each claim, we assume that the inequalities in the previous claims hold, including the claims in the proof of Theorem 1.

Claim 5. *With failure probability $O(n^{-5})$,*

$$\frac{c^2\Delta}{64} \leq \Delta^* \leq 2\Delta.$$

Proof. We know from the proof of Theorem 1 that, with failure probability at most $O(n^{-5})$, for each $i \geq L - \log n$ there is some colour class V_j such that $\mu_j(Q_1^{(i)}, Q_2^{(i)}) \geq c^2|S_i|/32$. Thus, with $i = L - \log n$, if $v \in V_j \cap S_{i+1}$ then $\mathbb{E}\lambda(v, Q_2^{(i)}, Q_1^{(i)}) \geq c^2\Delta|S_i|/32$. It follows by Chernoff's inequality that $\mathbb{P}\left(\lambda(v, Q_2^{(i)}, Q_1^{(i)}) < c^2\Delta/64\right) < 1/4$, and so (8) implies that with probability at least $1 - O(n^{-5})$ we have, for every $i \geq L - \log n$, at least $c|S_i|/2$ vertices $v \in V_j \cap S_{i+1}$ with $\lambda(v, Q_2^{(i)}, Q_1^{(i)}) \geq c^2\Delta|S_i|/64$.

On the other hand, every vertex $v \in S_{i+1}$ has $|\mathbb{E}\lambda(v, Q_1^{(i)}, Q_2^{(i)})| \leq \Delta|S_i|$, and so by (7) we have

$$\mathbb{P}\left(|\lambda(v, Q_1^{(i)}, Q_2^{(i)})| > 2\Delta|S_i|\right) < \exp(-\Delta|S_i|) = O(n^{-5}).$$

It follows that with probability $1 - O(n^{-5})$, we have, for all $i \geq L - \log n$, fewer than $c|S_{i+1}|/2$ vertices $v \in S_{i+1}$ with $|\lambda(v, Q_1^{(i)}, Q_2^{(i)})| \geq 2\Delta|S_i|$.

The inequality now follows immediately. □

The next claim shows that, at each stage of Algorithm A, if the bias $|\mu_j(Q_1^{(i)}, Q_2^{(i)})|$ is not very small then either it is very large (so colour j is very biased), or it becomes very large at the next stage, pushing up the number of vertices with large imbalance at the stage after. (Note that once a colour class becomes very imbalanced, this persists by Claim 4.)

Claim 6. *Consider Algorithm A. With probability* $1 - O(n^{-6})$*, for* $i < L$ *and each colour* j*, if*

$$|\mu_j(Q_1^{(i)}, Q_2^{(i)})| \geq c^2|Q_1^{(i)}|/(\log n)^{1/10}$$

then

$$|\mu_j(Q_i^{(i+1)}, Q_2^{(i+1)})| \geq c^2|Q_1^{(i+1)}|/32.$$

Proof. If $\mu_j(Q_1^{(i)}, Q_2^{(i)}) \geq c^2|Q_1^{(i)}|/(\log n)^{1/10}$ then we argue as in the proof of Claim 4, except that the inequality $\mu \geq c^2|S_i|/32$ is replaced by $\mu \geq c^2|Q_1^{(i)}|/(\log n)^{1/10}$, which by (15) is at least $c^3|S_i|/6(\log n)^{1/10}$. Modifying (35)-(36) accordingly, we see that vertices in $V_j \cap S_{i+1}$ are pushed towards R_{i+1} with failure probability

$$\exp(-c^2(\log n)^{3/10}) \leq \exp(-100(\log n)^{1/20}).$$

Now (7) and (14) imply that

$$\mathbb{P}(|V_j \cap B_{i+1}| > c^3|S_i|/1000) = O(n^{-6}).$$

It follows from (15) and (16) that $\mu_j(Q_1^{(i+1)}, Q_2^{(i+1)}) \geq c^2|Q_1^{(i+1)}|/32$.

A symmetric argument deals with the case when $\mu_j(Q_1^{(i)}, Q_2^{(i)})$ is negative. □

Claim 7. *Let* $i = i^*$*. With probability* $1 - O(n^{-4})$*, after the backtracking step we have*

$$|\mu_j(Q_1^{(i)}, Q_2^{(i)})| \geq c^2|S_i|/32$$

for every colour j*.*

Proof. Note first that the algorithm cannot report failure in Step 3, by our choice of c, so i^* is well-defined. Now suppose we have just modified R'_{i^*} and B'_{i^*} to obtain $Q_1^{(i)}$ and $Q_2^{(i)}$. We show that with failure probability $O(n^{-5})$ we have $|\mu_j(Q_1^{(i)}, Q_2^{(i)})| > c^2|S_i|/32$ for each colour class.

Given j, we may assume that $\mu_j(Q_1^{(i-1)}, Q_2^{(i-1)}) \geq 0$. (The negative case is identical.)

If $\mu_j(Q_1^{(i-1)}, Q_2^{(i-1)}) > c^2|Q_1^{(i-1)}|/32$ then, for $v \in S_i \cap V_j$, modifying (35) to bound $\mathbb{P}(Y \leq 3\Delta\mu/5)$, it follows from the proof of Claim 4 that (17) also holds for $\mathbb{P}(v \notin R_{i+1}$ or v has small imbalance), and hence the second inequality in Clam 4 still holds.

On the other hand, if $\mu_j(Q_1^{(i-1)}, Q_2^{(i-1)}) < c^2|Q_1^{(i-1)}|/(\log n)^{1/10}$ then, for $v \in S_i \cap V_j$,

$$\mathbb{E}\lambda(v, Q_1^{(i-1)}, Q_2^{(i-1)}) \leq c^2\Delta|Q_1^{(i-1)}|/(\log n)^{1/10} \leq 64\Delta^*|Q_1^{(i-1)}|/(\log n)^{1/10}.$$

It follows from Chernoff's inequality and (37) that, with failure probability $O(n^{-6})$, at most $c^3|S_i|/32$ vertices in $|V_j \cap S_i|$ have strong imbalance, and (16) still holds for the modified sets $Q_l^{(i)}$, so $|\mu_j(Q_1^{(i)}, Q_2^{(i)})| \geq c^3|S_i|/32$.

Finally, if $\mu_j(Q_1^{(i-1)}, Q_2^{(i-1)})$ does not satisfy either of the inequalities in Claim 6, it follows from Claim 4, Claim 6 and Chernoff's inequality that, with failure probability $O(n^{-6})$, $l_{i^*+1} \geq l_{i^*-1} + c/2$ in step 3, which contradicts our choice of i^*. □

Claim 8. *With probability* $1 - O(n^{-5})$ *we end up with a perfect split.*

Proof. This follows easily from the preceding claims together with Claim 4. □

Acknowledgements. We would like to thank Svante Janson and an anonymous referee for their very helpful suggestions.

References

[1] N. Alon, M. Krivelevich and B. Sudakov, Finding a large hidden clique in a random graph, *in* Proc. SODA '98—Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco CA, January 25–27, 1998

[2] N. Alon and J. Spencer, *The probabilistic method*, Second edition, John Wiley and Sons, New York, 2000, xviii+301 pp.

[3] S. Arora, D. Karger, and M Karpinski, Polynomial time approximation schemes for dense instances of NP-hard problems, *in* 27th STOC, pages 284–293, 1995

[4] P. van Beck, Z. Wahr. verw. Geb. **23** (1972), 187–196

[5] A. Ben-Dor, R. Shamir, and Z. Yakhini, Clustering gene expression patterns, *Journal of Computational Biology* **6** (1999), 281–297

[6] R. B. Boppana. Eigenvalues and graph bisection: An average case analysis, *in* IEEE Symp. on Foundations of Computer Science, 280–285, 1987

[7] A. Broder, A.M. Frieze and E. Shamir, Finding hidden Hamiltonian cycles, *in* Proc. 23rd ACM Symp. on Theory of Computing, 1991, 182–189.

[8] T. N. Bui, S. Chaudhuri, F. T. Leighton, and M. Sipser, Graph Bisection Algorithms with Good Average Case Behavior, *Combinatorica* **7** (1987), 841–855

[9] T. Carson and R. Impagliazzo, Hill-climbing finds random planted bisections, *in* Symposium on Discrete Algorithms, 2001, 903–909

[10] A. Condon and R. M. Karp, Algorithms for graph partitioning on the planted partition model, *Random Structures and Algorithms* **18** (2001), no. 2, 116–140

[11] D. Coppersmith, D. Gamarnik,M. Hajiaghayi and G. B. Sorkin, Random MAX SAT, Random MAX CUT, and Their Phase Transitions, *Random Structures and Algorithms*, to appear

[12] M. Dyer and A. M. Frieze, The solution of some random NP-hard problems in polynomial expected time, *J. Algorithms* **10** (1989), 451–489

[13] U. Feige and J. Kilian, Heuristics for semirandom graph problems. Special issue on FOCS 98 (Palo Alto, CA), *J. Comput. System Sci.* **63** (2001), 639–671

[14] U. Feige and R. Krauthgamer, Finding and certifying a large hidden clique in a semi-random graph, *Random Structures and Algorithms* **16** (2000), 195–208

[15] A. Frieze and R. Kannan, The regularity lemma and approximation scheme for dense problems, *in* Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science, 1996, 12–20

[16] M.R. Garey, and D.S. Johnson, Computers and Intractability, W. H. Freeman and Company, New York, 1979

[17] M. R. Garey, D.S. Johnson and L. Stockmeyer, Some simplified NP-complete graph problems, *Theoretical Comp. Science* **1** (1976), 237–267

[18] J. Håstad, Some optimal inapproximability results, *in* STOC '97 (El Paso, TX), ACM, New York, 1–10

[19] S. Janson, On concentration of probability, *in* Contemporary Combinatorics, ed. B. Bollobás, Bolyai Society Mathematical Studies **10**, János Bolyai Mathematical Society, Budapest and Springer, Berlin, 2002, 289–301

[20] M. Jerrum and G. B. Sorkin, The Metropolis algorithm for graph bisection, *Discrete Appl. Math.* **82** (1998), 155–175

[21] A. Juels, *Topics in Black-box Combinatorial Function Optimization*, U.C. Berkeley Ph.D. Thesis Dissertation, 1996.

[22] A. Juels and M. Peinado, Hiding Cliques for Cryptographic Security, *in* Proceedings of the ninth annual ACM-SIAM Symposium on Discrete Algorithms, ACM Press, 1998

[23] V. Kann, S. Khanna, J. Lagergren and A. Panconesi, On the hardness of approximating Max k-Cut and its dual, *Chicago J. Theoret. Comput. Sci.* 1997, Article 2, 18 pp. (electronic)

[24] L. Kučera, Expected complexity of graph partitioning problems, Discrete Applied Mathematics **57** (1995), 193–212

[25] C.H. Papadimitriou and M. Yannakakis, Optimization, approximation, and complexity classes, *J. Comput. System Sci.* **43** (1991), 425–440

[26] V. V. Petrov, *Sums of independent random variables*, Translated from the Russian by A. A. Brown, Ergebnisse der Mathematik und ihrer Grenzgebiete, Band 82, Springer-Verlag, New York-Heidelberg, 1975, x+346pp.

[27] R. Shamir and D. Tsur, Improved algorithms for the random cluster model, SWAT 2002, Lecture Notes in Computer Science **2368** (2002), 230–239