Combinatorics, Probability and Computing (2007) 16, 515–552. © 2006 Cambridge University Press
doi:10.1017/S0963548306007917 First published online 14 August 2006 Printed in the United Kingdom

Colouring Semirandom Graphs

AMIN COJA-OGHLAN*

Humboldt-Universität zu Berlin, Institut für Informatik, Unter den Linden 6, 10099 Berlin, Germany
(e-mail: coja@informatik.hu-berlin.de)

Received 10 January 2005; revised 10 March 2006

We study semirandom k-colourable graphs made up as follows. Partition the vertex set $V = \{1, \ldots, n\}$ randomly into k classes $V_1, \ldots, V_k$ of equal size and include each V_i–V_j-edge with probability p independently ($1 \leqslant i < j \leqslant k$) to obtain a graph G_0. Then, an adversary may add further V_i–V_j-edges ($i \neq j$) to G_0, thereby completing the semirandom graph $G = G^*_{n,p,k}$. We show that if $np \geqslant \max\{(1+\varepsilon)k \ln n, C_0 k^2\}$ for a certain constant $C_0 > 0$ and an arbitrarily small but constant $\varepsilon > 0$, an *optimal* colouring of $G^*_{n,p,k}$ can be found in polynomial time with high probability. Furthermore, if $np \geqslant C_0 \max\{k \ln n, k^2\}$, a k-colouring of $G^*_{n,p,k}$ can be computed in polynomial expected time. Moreover, an *optimal* colouring of $G^*_{n,p,k}$ can be computed in expected polynomial time if $k \leqslant \ln^{1/3} n$ and $np \geqslant C_0 k \ln n$. By contrast, it is NP-hard to k-colour $G^*_{n,p,k}$ w.h.p. if $np \leqslant (\frac{1}{2} - \varepsilon)k \ln(n/k)$.

1. Introduction

1.1. Graph colouring heuristics

In the *Graph Colouring Problem* we are given a graph $G = (V, E)$, and the goal is to colour the vertices V with as few colours as possible such that adjacent vertices receive distinct colours. The least number of colours so that there exists such a colouring is the *chromatic number* $\chi(G)$.

While the Graph Colouring Problem is of fundamental interest in theoretical computer science as well as in discrete mathematics, the problem is notoriously hard. Indeed, Feige and Kilian [15] proved that no polynomial time algorithm approximates $\chi(G)$ within a factor of $n^{1-\varepsilon}$ for all input graphs G, unless ZPP = NP; here $n = \#V$, and $\varepsilon > 0$ is an arbitrarily small constant. Furthermore, Khanna, Linial and Safra [27] showed that it is NP-hard to colour 3-colourable graphs with 4 colours.

Nevertheless, these hardness results merely provide evidence that for every polynomial time algorithm there are *some* 'hard' problem instances. Hence, the hardness results do not rule out

* Research supported by the Deutsche Forschungsgemeinschaft (grant DFG FOR 413/1-1).

the existence of good *graph colouring heuristics* that perform well on 'almost all instances' in some meaningful sense. Therefore, the goal of this paper is to analyse graph colouring heuristics rigorously within the framework of the algorithmic theory of random graphs (see [20] for some background).

Of course, in order to obtain rigorous results, we need to specify precisely what 'almost all instances' is supposed to mean. One possible answer is to consider the well-known *Erdős–Rényi model* $G_{n,p}$ of random graphs. The random graph $G_{n,p}$ has n vertices $V = \{1, \ldots, n\}$, and each of the $\binom{n}{2}$ possible edges is present with probability p independently. Bollobás [5] and Łuczak [32] determined the probable value of $\chi(G_{n,p})$: we have

$$\chi(G_{n,p}) \sim -\frac{n\ln(1-p)}{2\ln(np)} \quad \text{w.h.p. if } n^{-1} \ll p \leqslant 0.99. \tag{1.1}$$

(For small edge probabilities $p = O(1/n)$, Achlioptas and Naor [1] obtained more precise results.) We emphasize that (1.1) shows that the chromatic number $\chi(G_{n,p})$ is fairly 'high'. For if $np = \Omega(\ln n)$, then with probability $1 - o(1)$ as $n \to \infty$ the maximum degree of $G_{n,p}$ is $O(np)$ (see [6, Chapter 3]). Therefore, the chromatic number is $O(np)$, and (1.1) is just by an $O(\ln(np))$-factor smaller than this trivial upper bound.

In order to investigate graphs with a smaller chromatic number than (1.1), Kučera [30] suggested a random model $G_{n,p,k}$ that has an additional parameter k to control the chromatic number. The random graph $G_{n,p,k}$ is obtained as follows.

M1. Partition the vertex set $V = \{1, \ldots, n\}$ randomly into k classes $V_1, \ldots, V_k$ of equal cardinality (we assume that k divides n).

M2. Include every V_i–V_j-edge ($i \neq j$) with probability p independently of all others to obtain $G_0 = G_{n,p,k}$.

Thus, $V_1, \ldots, V_k$ is a k-colouring 'planted' in $G_{n,p,k}$, so that $\chi(G_{n,p,k}) \leqslant k$. We say that $G_{n,p,k}$ has some property $\mathcal{E}$ *with high probability* ('w.h.p.') if the probability that $\mathcal{E}$ holds tends to 1 as $n \to \infty$.

However, the $G_{n,p}$ and the $G_{n,p,k}$ model share a serious drawback: in both models the instances are purely random. As the theory of random graphs shows (see [25]), such instances have a very particular combinatorial structure. Therefore, designing heuristics for $G_{n,p}$ or $G_{n,p,k}$ yields heuristics for a *very special class* of graphs. Consequently, heuristics for purely random instances may lack 'robustness', as even minor changes in the structure of the input may deteriorate the algorithm's performance.

Therefore, Blum and Spencer [4] suggested a *semirandom* model $G^*_{n,p,k}$ that is closer to the worst case than $G_{n,p,k}$. The semirandom graph $G^*_{n,p,k}$ is obtained as follows. First, a random graph $G_0 = G_{n,p,k}$ is chosen via M1–M2; let $V_1, \ldots, V_k$ signify its planted k-colouring. Then, an adversary completes the problem instance as follows.

M3. The adversary may add to G_0 further V_i–V_j-edges ($i \neq j$) to obtain $G = G^*_{n,p,k}$.

Note that $V_1, \ldots, V_k$ remains a 'planted' k-colouring of $G^*_{n,p,k}$. Hence, $\chi(G^*_{n,p,k}) \leqslant k$.

Let $\mathcal{I}(G_0)$ signify the set of all graphs that can be obtained from $G_0 = G_{n,p,k}$ via M3. We say that $G^*_{n,p,k}$ has some property $\mathcal{E}$ *with high probability* ('w.h.p.') if $\mathcal{E}$ holds with probability $1 - o(1)$ as $n \to \infty$ regardless of the adversary's decisions. That is,

$$\lim_{n\to\infty} \mathrm{P}[G_0 = G_{n,p,k} \text{ is such that } \mathcal{E} \text{ holds for all } G \in \mathcal{I}(G_0)] = 1.$$

In contrast to $G_{n,p,k}$, the semirandom graph $G^*_{n,p,k}$ does not *consist* of random edges, but *contains* some random edges. Therefore, $G^*_{n,p,k}$ models a somewhat more general type of instances. On the one hand, the adversary can alter certain 'statistical' properties of $G_0 = G_{n,p,k}$. For example, the adversary can change the distribution of the vertex degrees or add 'dense spots' to the graph, thereby changing also spectral properties. On the other hand, the adversary is just allowed to add edges that 'point towards' the hidden colouring $V_1, \ldots, V_k$. Thus, intuitively the adversary just seems to make the problem 'easier'. Therefore, it appears natural to require that a 'robust' heuristic should not get confused by the adversary's actions. In other words, the $G^*_{n,p,k}$ model discriminates between heuristics that are robust enough to withstand such an adversarial 'help', and heuristics that are not.

Let us discuss the difference between $G_{n,p,k}$ and $G^*_{n,p,k}$ with a concrete example. Alon and Kahale [2] suggested a spectral heuristic that k-colours $G_{n,p,k}$ w.h.p. if k is fixed and $p > C_k/n$ for a certain constant $C_k > 0$. Given an input instance $G_0 = G_{n,p,k}$, the heuristic first removes all vertices of degree greater than $5np$, thereby obtaining a graph G'_0. Then, the heuristic computes the $k-1$ eigenvectors of the adjacency $A(G'_0)$ of G'_0 with the smallest eigenvalues. These eigenvectors yield a partition of G'_0 that is 'close' to the planted colouring of G_0 w.h.p. Finally, in order to obtain an actual k-colouring of G_0, the heuristic improves this partition via various combinatorial techniques.

However, this spectral approach breaks down on the $G^*_{n,p,k}$ model. Let us assume for concreteness that $k = 3$, and that $C_3 \leqslant np = O(1)$. Then w.h.p. each of the planted colour classes V_1, V_2, V_3 of G_0 contains $\Omega(n)$ isolated vertices. Hence, w.h.p. the adversary can pick disjoint sets $A_1, A_2 \subset V_1$, $B_1, B_2 \subset V_2$ of isolated vertices such that $\#A_i = \#B_i = 2np/3$. Then, the adversary adds all A_i–B_i-edges to G_0 to obtain a graph G. Thus, in G both (A_1, B_1) and (A_2, B_2) are bipartite cliques. Let G' be the graph obtained by removing all vertices of degree $> 5np$ from G, and let A be the adjacency matrix of G'. Then similar spectral arguments as in [2] show that the two eigenvectors of A with the smallest eigenvalues just represent the bipartite cliques (A_1, B_1), (A_2, B_2), but do not encode any useful information to 3-colour G. Thus, the adversary can jumble up the spectrum of G_0 to render the spectral approach useless. (A similar construction shows that on $G^*_{n,p,k}$ the spectral approach breaks down also for larger values of p – say, $np = n^{o(1)}$.)

1.2. Results

The goal of this paper is to investigate heuristics for colouring $G^*_{n,p,k}$. First, we present a simple heuristic that computes an *optimal* colouring of $G^*_{n,p,k}$ in polynomial time w.h.p. In addition, we suggest heuristics for colouring $G^*_{n,p,k}$ in polynomial expected time. We will compare these results with previous work in Section 1.3.

1.2.1. Colouring $G^*_{n,p,k}$ optimally. While $G^*_{n,p,k}$ is always k-colourable, it might happen that the chromatic number is actually smaller than k. Therefore, we say that a heuristic $\mathcal{A}$ *colours $G^*_{n,p,k}$ optimally w.h.p.* if the following two conditions are satisfied.

Correctness. For all input graphs G the algorithm $\mathcal{A}$ either outputs an optimal colouring or 'fail'.
Completeness. On input $G^*_{n,p,k}$, the output is an optimal colouring w.h.p.

https://doi.org/10.1017/S0963548306007917 Published online by Cambridge University Press

Thus, we require $\mathcal{A}$ not only to find a colouring of the input graph, but also to compute a matching *lower* bound on the chromatic number. In other words, $\mathcal{A}$ is supposed to *certify* that its output is an optimal colouring.

Theorem 1.1. *There is a polynomial time algorithm* `Colour` *such that the following holds. Let $\varepsilon > 0$ be arbitrarily small but constant. Moreover, suppose that $k = k(n)$ and $p = p(n)$ are such that*

$$np \geqslant \max\{(1+\varepsilon)k \ln n, C_0 k^2\} \text{ for a certain constant } C_0 > 0. \tag{1.2}$$

Then `Colour` *colours $G^*_{n,p,k}$ optimally w.h.p.*

Note that for $k = o(\ln n)$ – hence in particular for constant k – the assumption in Theorem 1.1 just reads $np \geqslant (1+\varepsilon)k \ln(n)$. `Colour` employs a semidefinite programming ('SDP') relaxation $\bar{\vartheta}_2$ of the chromatic number (we will recall the definition in Section 2). The basic observation is that on $G = G^*_{n,p,k}$ w.h.p. all optimal fractional solutions to $\bar{\vartheta}_2$ are *integral*, *i.e.*, encode actual colourings of G.

The algorithm `Colour` can be considered as a 'more robust' version of the spectral heuristic of Alon and Kahale [2]. More precisely, `Colour` can cope with the semirandom model $G^*_{n,p,k}$, because we replace the spectral techniques by SDP techniques. Nevertheless, the *proof* that all optimal fractional solutions to the SDP $\bar{\vartheta}_2$ are integral w.h.p. relies on SDP duality and extends the spectral considerations of Alon and Kahale. Extending the spectral techniques to the semirandom model was posed as an open problem by Frieze and McDiarmid [20, Research Problem 19].

The following hardness result complements Theorem 1.1.

Theorem 1.2. *Let $3 \leqslant k \leqslant n^{1/2}$. There is no polynomial time algorithm that in the case*

$$np \leqslant (1-\varepsilon)\frac{k}{2}\ln(n/k) \tag{1.3}$$

k-colours $G^*_{n,p,k}$ w.h.p., unless NP $\subset$ RP.

If $k = o(\ln n)$, then conditions (1.2) and (1.3) differ just by a factor of 2.

1.2.2. Colouring $G^*_{n,p,k}$ in expected polynomial time. Despite Theorem 1.2, can we push the positive result Theorem 1.1 any further? The algorithm `Colour` for Theorem 1.1 runs *always* in polynomial time and k-colours $G^*_{n,p,k}$ *with high probability*. One way to strengthen this result is to devise an algorithm that even k-colours *any* k-colourable input graph such that the *expected* running time over $G^*_{n,p,k}$ is polynomial. Here we define the *expected running time* of an algorithm $\mathcal{A}$ on input $G^*_{n,p,k}$ as

$$\sum_{G_0} \mathrm{P}(G_0 = G_{n,p,k}) \cdot \max_{G \in \mathcal{I}(G_0)} R_{\mathcal{A}}(G),$$

where $R_{\mathcal{A}}(G)$ denotes the running time of $\mathcal{A}$ on input G, and the sum ranges over all possible outcomes G_0 of $G_{n,p,k}$. The following theorem shows that there is a colouring algorithm with polynomial expected running time for almost the same range of the parameters as in Theorem 1.1.

Theorem 1.3. *Suppose that $k = k(n)$ and $p = p(n)$ are such that*

$$np \geqslant C_0 \max\{k \cdot \ln n, k^2\} \text{ for a certain constant } C_0 > 0. \tag{1.4}$$

There is an algorithm `ExpColour` *that k-colours any k-colourable input graph and that applied to $G^*_{n,p,k}$ has polynomial expected running time.*

To achieve the algorithm `ExpColour` with polynomial expected running time, we need to refine the heuristic `Colour` significantly. Indeed, while `Colour` may just 'give up' if the input lacks certain 'typical' properties of $G^*_{n,p,k}$, `ExpColour` must be able to handle *all* k-colourable input graphs. Hence, if we imagine `ExpColour`'s quest for a k-colouring as a search tree, then this search tree can be of polynomial or of exponential size, or anything in between. Therefore, in order to guarantee a polynomial expected running time, we need to extend `Colour` and its analysis in two respects.

- We need to improve the algorithm so that the size of the search tree is distributed 'smoothly' such that it is small on average. Loosely speaking, this means that `ExpColour` needs to cope with minor 'atypical defects' in the input instance in such a way that the running time scales 'reasonably' as a function of the size of the 'defect'.
- We need to invent methods to analyse the average size of the search tree. In particular, we need to quantify how 'typical' or 'atypical' a certain input graph is, in terms of the $G^*_{n,p,k}$ model.

For $k = o(\ln n)$ Theorem 1.2 shows that the bound (1.4) on p is best possible up to the precise value of C_0. However, in contrast to `Colour`, `ExpColour` does not certify the optimality of the obtained colouring. Nevertheless, at least for $k \leqslant \ln^{1/3} n$ (and hence in particular for constant k), it is easy to modify `ExpColour` to obtain an algorithm that certifies the optimality of its output.

Theorem 1.4. *Suppose that $k = k(n)$ and $p = p(n)$ are such that*

$$np \geqslant C_0 k \cdot \ln n \text{ for a certain constant } C_0 > 0, \text{ and } k \leqslant \ln^{1/3} n. \tag{1.5}$$

There is an algorithm `OptColour` *that colours any input graph optimally and that applied to $G^*_{n,p,k}$ has polynomial expected running time.*

1.3. Related work

Blum and Spencer [4] were the first to study the $G^*_{n,p,k}$ model. They showed that a k-colouring of $G^*_{n,p,k}$ can be found in polynomial time w.h.p. if k is constant and

$$np \geqslant n^{\alpha_k + \varepsilon}, \qquad \text{where } \alpha_k = \frac{k^2 - k - 2}{k^2 + k - 2} \tag{1.6}$$

and $\varepsilon > 0$ is an arbitrarily small constant. This colouring heuristic is purely combinatorial.

Feige and Kilian [16] suggested the strongest previous heuristic for colouring $G^*_{n,p,k}$. The heuristic finds a k-colouring in polynomial time w.h.p. if k is constant and $np \geqslant (1 + \varepsilon)k \ln n$. Note that for constant k this assumption is identical to (1.2). In order to k-colour $G^*_{n,p,k}$, the heuristic tries to recover the classes of the planted k-colouring one by one. To recover a colour class, the heuristic combines SDP techniques for approximating the independence number from Alon and Kahale [3] with the random hyperplane rounding technique from Goemans and Williamson [22]. These SDP techniques are needed to obtain an initial partition of the input graph

https://doi.org/10.1017/S0963548306007917 Published online by Cambridge University Press

that consists of relatively 'sparse' sets. Then, the heuristic makes use of matching techniques and expansion properties of $G^*_{n,p,k}$ to extract the colour class from the initial partition.

Theorem 1.1 improves on the result of Feige and Kilian in the following respects.

- It is not clear whether the heuristic in [16] is applicable when k grows as a function of n (say, $k \gg \ln n$), because the analysis of the SDP rounding techniques in [16] requires that the initial partition consists of $\exp(\Omega(k))$ classes to guarantee that the classes of the partition are sparse enough. On the other hand, choosing $p = 1/2$ we can make k as large as $\Omega(\sqrt{n})$ in Theorem 1.1.
- The algorithm `Colour` is much simpler. For instance, it needs to solve an SDP only once, whereas the heuristic of Feige and Kilian requires several SDP computations. (Nonetheless, the techniques in [16] apply to further problems that we do not address in this paper.)
- Instead of just producing a k-colouring of $G = G^*_{n,p,k}$ w.h.p., `Colour` also provides a *certificate* that its output is indeed optimal.

In addition, Feige and Kilian [16] proved that no polynomial time algorithm k-colours $G^*_{n,p,k}$ w.h.p. if $np \leqslant (1-\varepsilon)\ln n$, unless NP $\subset$ RP. Theorem 1.2 improves this result by a factor of $k/2$, although the proof uses a similar idea.

Theorems 1.3 and 1.4 also improve on a colouring algorithm in [11], which is based on similar techniques as the algorithm of Feige and Kilian [16]. The algorithm k-colours any k-colourable input graph, and the expected running time on $G^*_{n,p,k}$ is $n^{\Theta(k)}$, provided that $np \gg k \ln n$. Hence, the running time becomes superpolynomial if $k = k(n)$ grows as a function of n. By contrast, the expected running time of `ExpColour` is polynomial in both n and k (see Theorem 1.3). Furthermore, in contrast to the algorithm `OptColour` (see Theorem 1.4), even for constant k the colouring algorithm in [11] does not certify the optimality of its output.

Building on [35], Subramanian [34] gave a heuristic for colouring $G^*_{n,p,k}$ optimally in polynomial expected time for constant values of k under the assumption (1.6). The heuristic is purely combinatorial, and the certificate of optimality is just a clique of size k w.h.p. Theorem 1.4 extends this result to significantly smaller values of p. In fact, for small edge probabilities $p = C_0 k \ln n$ as in Theorem 1.4, the clique number of $G_{n,p,k}$ is 3 w.h.p. Hence, $G^*_{n,p,k}$ has no clique of size k (unless the adversary includes one) that yields a certificate of optimality.

With respect to colouring $G_{n,p,k}$, Kučera [30] presented a simple heuristic that for $k = O(\sqrt{n/\ln n})$ and $p = 1/2$ recovers the planted k-colouring of $G_{n,p,k}$ w.h.p. Note that Theorem 1.1 provides a slightly stronger result: `Colour` colours $G_{n,p,k}$ optimally if $p = 1/2$ and $k \leqslant c\sqrt{n}$ for a certain constant $c > 0$.

Dyer and Frieze [14] showed that an optimal colouring of $G_{n,p,k}$ can be found in polynomial expected time if $p = \Omega(1)$ remains bounded away from 0 as $n \to \infty$. Moreover, the best previous heuristic for colouring $G_{n,p,k}$ in polynomial expected time is due to Subramanian [34]. The heuristic is combinatorial and colours $G_{n,p,k}$ optimally in polynomial expected time if k is constant and

$$np \geqslant n^{\gamma(k)+\varepsilon}, \quad \text{where } \gamma(k) = \frac{k^2 - 3k + 2}{k^2 - k + 2}.$$

Theorem 1.4 provides a colouring heuristic that also applies to significantly smaller values of p. Extending Subramanian's result to smaller values of p was also mentioned as an open problem in the survey of Krivelevich [28, Section 7].

Some heuristics for random instances of more general partitioning problems also entail results on colouring $G_{n,p,k}$. For instance, the heuristic of Subramanian and Veni Madhavan [36], which is based on breadth first search, k-colours $G_{n,p,k}$ in polynomial time w.h.p. if k is constant and $np \geqslant \exp(C\sqrt{\ln n})$ for a certain constant $C > 0$. Moreover, McSherry's spectral heuristic [33] finds a k-colouring in polynomial time w.h.p. if k is constant and $np \gg \ln^3 n$. Finally, a randomized linear time partitioning heuristic of Bollobás and Scott [7] recovers the hidden colouring w.h.p. if $np \geqslant Ck^2 \ln n$ for a certain constant $C > 0$. Indeed, Bollobás and Scott conjecture that their heuristic can also handle the semirandom graph $G^*_{n,p,k}$. Some further references on colouring random and semirandom graphs can be found in the survey [28].

There are two recent papers that build upon the techniques developed in the present work. Complementing the work of Alon and Kahale [2], Böttcher [9] presented an algorithm that k-colours the random graph $G_{n,p,k}$ in polynomial expected time if $np \geqslant Ck^2$ for a certain constant $C > 0$. The algorithm makes use of the SDP techniques presented in Section 4 and an appropriate extension of the combinatorial methods from [2].

Furthermore, building on the present work and [9], Krivelevich and Vilenchik [29] suggested an algorithm that can cope with *sparse* semirandom instances $G^*_{n,p,k}$, $np < (1-\varepsilon)k \ln n$. Note that this range of the parameters is not covered by Theorems 1.1, 1.3 and 1.4; in fact, Theorem 1.2 shows that actually k-colouring $G^*_{n,p,k}$ is hard if $np < (1-\varepsilon)\frac{k}{2}\ln n$. Nonetheless, the algorithm in [29] k-colours $G^*_{n,p,k}$ in time $(1+\exp(-\Omega(np/k)))^n \cdot n^{O(1)}$ w.h.p. even if $np < (1-\varepsilon)k \ln n$. While this may be superpolynomial, the point is that this bound on the running time is in general considerably better than the running time of worst-case exponential time algorithms for k-colouring. In addition, Krivelevich and Vilenchik considered a modification of the $G^*_{n,p,k}$ model where the adversary may only add edges between vertices in some canonically defined set $S \subset V$, and showed that in this case a k-colouring can be found in polynomial time w.h.p.

1.4. Techniques and outline

The algorithms `Colour`, `ExpColour` and `OptColour` for Theorems 1.1, 1.3, and 1.4 make use of different techniques than the previous algorithms for colouring $G_{n,p,k}$ and $G^*_{n,p,k}$. For instance, `Colour` relies on a direct analysis of the optimal solutions to the SDP relaxation $\bar{\vartheta}_2$ on $G^*_{n,p,k}$ (see Section 2 for the definition of $\bar{\vartheta}_2$). More precisely, we show that all optimal fractional solutions are in fact integral w.h.p., *i.e.*, correspond to k-colourings of $G^*_{n,p,k}$. While the algorithm for colouring semirandom graphs in [16] is also based on SDP techniques (see Section 1.3), `Colour` is rather different: the analysis of `Colour` shows that there is a single SDP that captures the problem completely.

The techniques in the analysis of `Colour` extend previous work of Boppana [8] and Feige and Kilian [16] on the Min Bisection problem. More precisely, in [8] it was shown that all optimal fraction solutions to an SDP relaxation of Min Bisection correspond to actual bisections w.h.p. on certain random instances; this analysis was extended in [16] to semirandom models. Nevertheless, the analysis of $\bar{\vartheta}_2$ on the $G^*_{n,p,k}$ model turns out to be significantly more involved than the analyses for Min Bisection in [8, 16]. One reason is that while in the Min Bisection problem the goal is to recover *two* classes, the number $k = k(n)$ of colour classes in the $G^*_{n,p,k}$ model may actually grow as a function of n. A further reference is the work of Feige and Krauthgamer [17] on semirandom instances of the MAX CLIQUE problem; the heuristic is based on the integrality of optimal fractional solutions to an SDP relaxation of the clique number.

In order to obtain the heuristic `ExpColour` with polynomial expected running time, we need to refine the investigation of $\bar{\vartheta}_2$ on $G^*_{n,p,k}$. While `Colour` relies on the fact that all fractional solutions are perfectly integral w.h.p., `ExpColour` is based on the observation that with probability extremely close to 1 all fractional solutions are at least 'not too far' from being integral. To prove this statement, we invoke results from Coja-Oghlan, Moore and Sanwalani [13] on semidefinite relaxations of MAX k-CUT on the Erdős–Rényi model $G_{n,p}$. In addition, to extract the colouring from the fractional solution, `ExpColour` employs network flow techniques from [12], which extend matching techniques from [16]. Finally, `OptColour` combines `ExpColour` with a technique for computing a lower bound on $\chi(G^*_{n,p,k})$.

The heuristic `Colour` and its analysis are the content of Section 3. Moreover, we present `ExpColour` in Section 4. Then, in Section 5 we modify `ExpColour` to obtain the algorithm `OptColour` for Theorem 1.4. Section 6 is devoted to the proof of Theorem 1.2. Finally, Section 7 contains the proofs of some technical lemmas.

There are various constants involved in the analyses of the algorithms. Most of the constants are somewhat arbitrary and are only made explicit for concreteness; no attempt has been made to optimize these constants.

2. Preliminaries

2.1. Notation

Throughout, we let $V = \{1, \ldots, n\}$. Moreover, if X is a set, then we let $\delta_{x,X} = 1$ if $x \in X$ and $\delta_{x,X} = 0$ otherwise.

If G is a graph, then we let $V(G)$ denote the vertex set and $E(G)$ the edge set of G. For a set $A \subset V(G)$, $N(A) = N_G(A) = \{w \in V(G) : \exists v \in A : \{v, w\} \in E(G)\}$ signifies the neighbourhood of A. Moreover, $\bar{N}(A) = \bar{N}_G(A) = V(G) \setminus N_G(A)$ denotes the non-neighbourhood. Furthermore, by $G[A]$ we denote the subgraph of G induced on A. If $B \subset V(G)$ is a further set, then we let $e(A, B) = e_G(A, B)$ be the number of A–B-edges, *i.e.*,

$$e(A, B) = e_G(A, B) = \#\{\{v, w\} \in E(G) : v \in A, w \in B\}.$$

In addition, we let $e(A) = e_G(A) = e_G(A, A)$.

If $G = G^*_{n,p,k}$, then we let $G_0 = G_{n,p,k}$ denote the random graph from which G has been obtained via M3. Moreover, we let $V_1, \ldots, V_k$ denote the planted colour classes of G and G_0. If $U \subset \{1, \ldots, k\}$, then we let $V_U = \bigcup_{u \in U} V_u$.

The scalar product of two vectors $\xi, \eta \in \mathbb{R}^n$ is denoted by $\langle \xi, \eta \rangle$. Moreover, $\|\xi\| = \langle \xi, \xi \rangle^{1/2}$ signifies the L^2-norm. We let $\mathbf{1} = \mathbf{1}_n \in \mathbb{R}^n$ denote the vector with all entries equal to 1. In addition, if X a set and $A \subset X$, then $\mathbf{1}_A = (e_x)_{x \in X}$ denotes the vector with entries $e_x = 1$ if $x \in A$ and $e_x = 0$ if $x \in X \setminus A$. If $\xi \in \mathbb{R}^n$ is a vector, then $\operatorname{diag}(\xi)$ signifies the $n \times n$ matrix with diagonal ξ whose off-diagonal entries are 0.

The eigenvalues of a real symmetric $n \times n$ matrix A are denoted by $\lambda_1(A) \geqslant \cdots \geqslant \lambda_n(A)$. If A, B are symmetric $n \times n$ matrices, then we write $A \geqslant B$ if $\lambda_n(A - B) \geqslant 0$. Recall that A is positive semidefinite if $\lambda_n(A) \geqslant 0$, *i.e.*, $A \geqslant 0$. Furthermore, for an $n \times n$ matrix M we let $\|M\| = \max_{\xi \in \mathbb{R}^n, \|\xi\|=1} \|M\xi\|$. In addition, $\operatorname{diag}(M) \in \mathbb{R}^n$ is the vector consisting of the diagonal

https://doi.org/10.1017/S0963548306007917 Published online by Cambridge University Press

entries of M. By $\mathbf{J}$ we denote a matrix with all entries equal to 1 (of any size). Moreover, $\mathbf{E} = \mathrm{diag}(\mathbf{1})$ signifies the matrix with ones on the diagonal and off-diagonal entries equal to 0.

We shall mainly be interested in matrices associated with graphs. The adjacency matrix of a graph G is denoted by $A(G)$. In addition, $L(G) = \mathrm{diag}(A(G)\mathbf{1}) - A(G)$ signifies the Laplacian.

2.2. An SDP relaxation of the chromatic number

The colouring heuristics rely on a semidefinite programming ('SDP') relaxation $\bar{\vartheta}_2$ of the chromatic number. The semidefinite program was first defined by Goemans and Kleinberg [21] and was further studied by Charikar [10] and Szegedy [37]. Following Charikar, we define $\bar{\vartheta}_2$ in terms of vector colourings; this approach is related to the work of Karger, Motwani and Sudan [26].

Let $G = (V, E)$ be a graph with vertex set $V = \{1, \ldots, n\}$. Let $(v_1, \ldots, v_n)$ be an n-tuple of unit vectors in $\mathbb{R}^n$, and let $k > 1$. We call $(v_1, \ldots, v_n)$ a *rigid vector k-colouring* if

$$\langle v_i, v_j\rangle = (1-k)^{-1} \quad \text{for all } \{i, j\} \in E, \text{ and } \langle v_i, v_j\rangle \geqslant (1-k)^{-1} \quad \text{for all } \{i, j\} \notin E.$$

Now, we define $\bar{\vartheta}_2(G) = \inf\{k > 1 : G \text{ admits a rigid vector } k\text{-colouring}\}$. Since $\bar{\vartheta}_2(G)$ can be stated as a semidefinite program, the number $\bar{\vartheta}_2(G)$ and a rigid vector $\bar{\vartheta}_2(G)$-colouring can be computed in polynomial time within a tiny numerical error, *e.g.*, via the ellipsoid method (see [23, 37]).

Furthermore, we have $\bar{\vartheta}_2(G) \leqslant \chi(G)$. For assume that G is k-colourable, and let $V_1, \ldots, V_k$ be a partition of V into k independent sets. Moreover, let $(\xi_1, \ldots, \xi_k)$ be a family of unit vectors in $\mathbb{R}^{k-1}$ such that $\langle \xi_i, \xi_j\rangle = -(k-1)^{-1}$ if $i \neq j$; such a family can be constructed inductively and it is unique up to an orthogonal transformation. Let $v_i = \xi_j$ for all $i \in V_j$. Then $(v_i)_{i\in V}$ is a rigid vector k-colouring of G, whence $\bar{\vartheta}_2(G) \leqslant k$. Indeed, $\bar{\vartheta}_2$ is a tighter relaxation of χ than both the vector chromatic number from [26] and the Lovász number $\vartheta(\bar{G})$ (see [23, 37]).

Let $A = A(G) = (a_{ij})_{i,j\in V}$ be the adjacency matrix of G. Moreover, let $L = L(G)$ signify the Laplacian. Let $k \geqslant 2$. In addition to $\bar{\vartheta}_2(G)$, we also need the following SDP from Frieze and Jerrum [19]:

$$\begin{aligned} \mathrm{SDP}_k(G) \;=\; \max \sum_{1\leqslant i<j\leqslant n} a_{ij} \cdot \frac{k-1}{k}(1 - \langle v_i, v_j\rangle) \qquad (2.1) \\ \text{s.t. } \|v_i\| = 1 \text{ for } i = 1, \ldots, n, \\ \langle v_i, v_j\rangle \geqslant (1-k)^{-1} \text{ for all } 1 \leqslant i < j \leqslant n, \\ v_1, \ldots, v_n \in \mathbb{R}^n. \end{aligned}$$

If k is an integer, then $\mathrm{SDP}_k(G)$ is an upper bound on the weight of a MAX k-CUT of G. In particular, $\mathrm{SDP}_2(G)$ equals the MAX CUT relaxation of Goemans and Williamson [22].

An important property of SDP_k is that the semidefinite program is *monotone*:

$$\text{if } G' \text{ contains } G \text{ as a subgraph, then } \mathrm{SDP}_k(G') \geqslant \mathrm{SDP}_k(G). \qquad (2.2)$$

Furthermore, $\bar{\vartheta}_2(G)$ and $\mathrm{SDP}_k(G)$ are related as follows: if G has a rigid vector k-colouring $(v_i)_{i\in V}$, then $(v_i)_{i\in V}$ is a feasible solution to SDP_k with objective function value

$$\sum_{1\leqslant i<j\leqslant n} a_{ij} \cdot \frac{k-1}{k}(1 - \langle v_i, v_j\rangle) = \#E. \qquad (2.3)$$

https://doi.org/10.1017/S0963548306007917 Published online by Cambridge University Press

As trivially $\mathrm{SDP}_k(G) \leqslant \#E$, we conclude that $\mathrm{SDP}_k(G) = \#E$. Conversely, if $(v_i')_{i\in V}$ is a feasible solution to $\mathrm{SDP}_k(G)$ with objective function value $\#E$, then $(v_i')_{i\in V}$ is a rigid vector k-colouring.

To prove Theorem 1.3, we need the following result, which is an immediate consequence of [13, Theorems 3 and 4].

Lemma 2.1. *There exist constants $\zeta_0, \zeta_1 > 0$ such that the following holds. Suppose that $np \geqslant \zeta_0$. Then for all $k \geqslant 2$ we have* $\mathrm{P}\left[\mathrm{SDP}_k(G_{n,p}) \leqslant (1-k^{-1})\binom{n}{2}p + \zeta_1 n^{3/2}p^{1/2}\right] \geqslant 1 - \exp(-300n)$.

2.3. Eigenvalues of random matrices

The proof of Theorem 1.1 relies on estimates of the eigenvalues of $A(G_{n,p,k})$. In order to estimate these eigenvalues, we employ the following two results.

Lemma 2.2. *Suppose that $np \geqslant c_1 \ln n$ for a constant $c_1 > 0$. Then there exists a number $c_2 > 0$ that depends only on c_1 such that, with probability $\geqslant 1 - O(n^{-2}p^{-1})$, the random symmetric matrix $A = A(G_{n,p})$ enjoys the following property: $\forall \mathbf{1} \perp \eta \in \mathbb{R}^n : \|A\eta\| \leqslant c_2\sqrt{np}\cdot\|\eta\|$.*

Lemma 2.3. *Suppose that $np \geqslant c_1 \ln n$ for a constant $c_1 > 0$. Then there exists a number $c_2 > 0$ that depends only on c_1 such that, with probability $\geqslant 1 - O(n^{-2}p^{-1})$, the following holds. Let $A = (a_{ij})_{i,j=1,\dots,n}$ be a matrix whose entries are mutually independent random variables such that $a_{ii} = 0$ for all i and*

$$p = \mathrm{P}(a_{ij} = 1) = 1 - \mathrm{P}(a_{ij} = 0) \quad (i \neq j).$$

Then $\forall \mathbf{1} \perp \eta \in \mathbb{R}^n : \|A\eta\| \leqslant c_2\sqrt{np}\cdot\|\eta\|$.

Lemma 2.2 is implicit in Feige and Ofek [18], and Lemma 2.3 in Alon and Kahale [2].

2.4. Chernoff bounds

Assume that X is binomially distributed with parameters (n, p). Let $\mu = \mathrm{E}(X) = np$. We frequently need the following *Chernoff bounds* on the tails of X (see [25, Chapter 2] for proofs):

$$\mathrm{P}(X \geqslant \mu + t) \leqslant \exp\left(-\frac{t^2}{2(\mu + t/3)}\right),\ \mathrm{P}(X \leqslant \mu - t) \leqslant \exp\left(-\frac{t^2}{2\mu}\right) \quad (0 < t). \tag{2.4}$$

Moreover, letting $\phi(x) = (1+x)\ln(1-x) - x$ for $x > -1$, we have

$$\mathrm{P}(X \leqslant \mu - t) \leqslant \exp\left(-\mu\phi\left(\frac{-t}{\mu}\right)\right) \quad (0 < t < \mu). \tag{2.5}$$

3. A simple heuristic for finding an optimal colouring

3.1. Outline

We assume that (1.2) *is satisfied with a sufficiently large constant $C_0 > 0$, which will be specified implicitly in the analysis.* The algorithm `Colour` for Theorem 1.1 is shown in Figure 1.

https://doi.org/10.1017/S0963548306007917 Published online by Cambridge University Press

---

Algorithm 1. `Colour`(G)
Input: A graph $G = (V, E)$. *Output:* Either a $\chi(G)$-colouring of G or 'fail'.

1. Compute $\bar{\vartheta}_2(G)$ along with a rigid vector $\bar{\vartheta}_2(G)$-colouring $(x_v)_{v\in V}$ of G.
2. Let $H = (V, F)$ be the graph with edge set $F = \{\{v, w\} : \langle x_v, x_w\rangle \leqslant 0.995\}$. Apply the greedy colouring algorithm to H, and let $\mathcal{C}$ be the resulting colouring.
3. If $\mathcal{C}$ uses at most $\lceil\bar{\vartheta}_2(G)\rceil$ colours, then output $\mathcal{C}$ as a colouring of G. Otherwise, output 'fail'.

Figure 1. The algorithm `Colour`.

In summary, `Colour`(G) computes the rigid vector colouring $(x_v)_{v\in V}$. This can be done in polynomial time via semidefinite programming (see Section 2). Then, `Colour` constructs an auxiliary graph H in which two vertices v, w are adjacent if and only if their distance $\|x_v - x_w\|$ is at least 0.1, *i.e.*, if x_v and x_w are 'far apart'. To this graph H, `Colour` applies the simple greedy colouring algorithm. (Recall that the greedy algorithm just goes through the vertices $v = 1, \ldots, n$ and colours each v with the least colour in $\{1, \ldots, n\}$ not yet used by the neighbours of v.)

To show that `Colour` either finds an optimal colouring of the input graph G or outputs 'fail', note that the graph H constructed in step 2 contains G as a subgraph. For if $\{v, w\} \in E$, then $\langle x_v, x_w\rangle \leqslant 0$. Since $\chi(G) \geqslant \bar{\vartheta}_2(G)$, $\mathcal{C}$ is an optimal colouring of G if $\mathcal{C}$ uses at most $\lceil\bar{\vartheta}_2(G)\rceil$ colours.

Hence, to prove Theorem 1.1, it remains to show that `Colour`$(G = G^*_{n,p,k})$ outputs an optimal colouring w.h.p. Let $V_1, \ldots, V_k$ be the k-colouring planted in G. Directed by the proof that $\bar{\vartheta}_2(G) \leqslant \chi(G)$ (see Section 2), we call a rigid vector k-colouring $(x_v)_{v\in V}$ *integral* if there are vectors $(x^*_i)_{i=1,\ldots,k}$ such that $x_v = x^*_i$ for all $v \in V_i$, and $\langle x^*_i, x^*_j\rangle = (1-k)^{-1}$ for $i \neq j$. In other words, $(x_v)_{v\in V}$ is integral if and only if the rigid vector colouring maps each colour class onto a single point, and the angle between the points corresponding to V_i and V_j is $\cos^{-1}\left[(1-k)^{-1}\right]$ if $i \neq j$.

If the rigid vector colouring $(x_v)_{v\in V}$ computed in step 1 is integral, then the graph H constructed in step 2 is a complete k-partite graph with colour classes $V_1, \ldots, V_k$. That is, in H the sets $V_1, \ldots, V_k$ are independent, but each $v \in V_i$ is connected with all vertices in $V \setminus V_i$. Consequently, the greedy algorithm finds a k-colouring of H. Hence, if also $\bar{\vartheta}_2(G) = k$, then `Colour` finds and outputs an optimal colouring of G. Thus, the remaining task is to establish the following lemma.

Lemma 3.1. *Let $G = G^*_{n,p,k}$. With high probability we have $\bar{\vartheta}_2(G) = k$, and every rigid vector k-colouring of G is integral.*

To prove Lemma 3.1, we make use of the relationship between SDP_h and $\bar{\vartheta}_2$ (see Section 2). With respect to SDP_h, we prove the following in Section 3.2.

Lemma 3.2. *There is a constant $\zeta > 0$ such that $G = G^*_{n,p,k}$ enjoys the following property w.h.p.:*

$$\text{Let } G' \text{ be a graph obtained by adding an edge } \{v^*, w^*\} \text{ to } G, \text{ where } v^*, w^* \in V_i \text{ for some } i. \text{ Let } 2 < h \leqslant k. \text{ Then } \mathrm{SDP}_h(G') \leqslant \#E(G) - \zeta \cdot \frac{n^2 p}{hk} \cdot (k-h). \tag{3.1}$$

Proof of Lemma 3.1. To prove that $\bar{\vartheta}_2(G^*_{n,p,k}) = k$ w.h.p., let $G = G^*_{n,p,k}$, and assume that $\bar{\vartheta}_2(G) = h < k$. Let $(x_v)_{v\in V}$ be a rigid vector h-colouring of G. Then $(x_v)_{v\in V}$ is a feasible solution to SDP_h, whence $\mathrm{SDP}_h(G) = \#E(G)$ due to (2.3). However, by Lemma 3.2 and the monotonicity property (2.2) we have $\mathrm{SDP}_h(G) < \#E(G)$ w.h.p. Thus, $\bar{\vartheta}_2(G) = k$ w.h.p.

Finally, to show that any rigid vector k-colouring $(x_v)_{v\in V}$ of $G = G^*_{n,p,k}$ is integral w.h.p., suppose that G has the property stated in Lemma 3.2. Let $s, t \in V_i^*$, and let G' be the graph obtained from G by adding the edge $\{s, t\}$. Then we have

$$\begin{aligned}
\#E(G) &= \frac{k-1}{k}\left[\sum_{\{v,w\}\in E(G)} 1 - \langle x_v, x_w\rangle\right] \\
&\leqslant \frac{k-1}{k}\left[1 - \langle x_s, x_t\rangle + \sum_{\{v,w\}\in E(G)} 1 - \langle x_v, x_w\rangle\right] \\
&\leqslant \mathrm{SDP}_k(G') \overset{\text{Lemma 3.2}}{\leqslant} \#E(G).
\end{aligned}$$

Therefore, $\langle x_s, x_t\rangle = 1$, whence $x_s = x_t$, because x_s, x_t are unit vectors. Consequently, there are unit vectors x_i^* such that $x_v = x_i^*$ for all $v \in V_i$, $i = 1, \ldots, k$.

Furthermore, if $i \neq j$, then $e_G(V_i, V_j)$ is binomially distributed with mean $n^2k^{-2}p$. Hence, our assumption (1.2) and the Chernoff bound (2.4) entail that $e_G(V_i, V_j) > 0$ for all $i \neq j$ w.h.p. Thus, let $v \in V_i$, $w \in V_j$ be vertices such that $\{v, w\} \in E(G)$. Then $\langle x_i^*, x_j^*\rangle = \langle x_v, x_w\rangle = (1-k)^{-1}$. Hence, the rigid vector colouring $(x_v)_{v\in V}$ is in fact integral. □

3.2. Proof of Lemma 3.2

SDP duality provides a powerful tool for proving an upper bound on the optimal solution to a maximization problem such as SDP_h. Let $G = (V, E)$ be a graph. Then the *dual semidefinite program* of $\mathrm{SDP}_h(G)$ reads

$$\begin{aligned}
\mathrm{DSDP}_h(G) \;=\; \min\; & \frac{h-1}{2h}\sum_{i=1}^{n} y_{ii} - \frac{1}{2h}\sum_{i\neq j} y_{ij} \\
\text{s.t. } & L(G) \leqslant Y, \\
& y_{ij} \leqslant 0 \text{ for } i \neq j, \\
& Y = (y_{ij})_{i,j=1,\ldots,n} \text{ is a real symmetric } n \times n \text{ matrix}
\end{aligned}$$

(see Helmberg [24, Chapter 2] for a thorough treatment of SDP duality theory). By weak SDP duality (see [24, pp. 17–18]), we have $\mathrm{SDP}_h(G) \leqslant \mathrm{DSDP}_h(G)$. Observe that the set of feasible solutions Y to $\mathrm{DSDP}_h(G)$ is the same for all values of h.

To prove Lemma 3.2, we exhibit a feasible solution to $\mathrm{DSDP}_h(G)$ for which the desired objective function value claimed in (3.1) is attained. The construction makes use of Lemmas 2.2 and 2.3. Let us first consider a random k-colourable graph $G = G_{n,p,k}$ with planted k-colouring

$V_1, \ldots, V_k$. As permuting the vertices does not affect the semidefinite program, we may assume that

$$V_i = \left\{ \frac{(i-1)n}{k} + 1, \ldots, \frac{in}{k} \right\} \quad (i = 1, \ldots, k). \tag{3.2}$$

Let G' be the graph obtained from G by adding an edge $\{v^*, w^*\}$, where $v^*, w^* \in V_{i_0}$ for some i_0. Let $A = A(G)$, $L' = L(G')$, $B = L' - L(G)$. For $v \in V$, let d_v be the degree of v in G, and let

$$d_v^{(i)} = e_G(v, V_i), \quad \bar{d} = \frac{1}{(k-1)n} \sum_{v \in V} d_v.$$

Let $i(V_a) = (a-1)\frac{n}{k} + i$ denote the ith vertex in V_a. Moreover, set

$$d_{\min} = \min_{i,j,a \neq b} \frac{n \cdot d_{i(V_a)}^{(b)} \cdot d_{j(V_b)}^{(a)}}{k \cdot e_G(V_a, V_b)}. \tag{3.3}$$

Further, we define a family of $\frac{n}{k} \times \frac{n}{k}$-matrices Y'_{ab} as follows: we let $Y'_{aa} = 0$ for $a = 1, \ldots, k$, and for $1 \leqslant a, b \leqslant k$, $a \neq b$, we let

$$Y'_{ab} = \left[\frac{k}{n} d_{\min} - \frac{d_{i(V_a)}^{(b)} \cdot d_{j(V_b)}^{(a)}}{e_G(V_a, V_b)} \right]_{i,j=1,\ldots,n/k}.$$

In addition, we let $Y' = (Y'_{ab})_{a,b=1,\ldots,k}$ be the $n \times n$ matrix comprising the blocks Y'_{ab}. Further, we let $y' = (d_v + d_{\min})_{v \in V} \in \mathbb{R}^n$, and finally $Y = Y' + \operatorname{diag}(y')$. Then Y is a real symmetric $n \times n$ matrix, and the definition (3.3) of $d_{\min}$ ensures that all off-diagonal entries of Y are $\leqslant 0$.

We claim that Y is a feasible solution to DSDP_h w.h.p. Thus, we need to show that $L' \leqslant Y$ w.h.p. Since $L' - Y = -(A - B + Y') - d_{\min}\mathbf{E}$, it suffices to prove that $\lambda_n(A - B + Y') \geqslant -d_{\min}$ w.h.p. As a first step, we shall exhibit a subspace $K \subset \mathbb{R}^n$ generated by eigenvectors of $A - B + Y'$ that correspond to the planted colouring $V_1, \ldots, V_k$. To this end, we note that

$$Y_{ab}\mathbf{1} = Y'_{ab}\mathbf{1} = \left[d_{\min} - \sum_{j=1}^{n/k} \frac{d_{i(V_a)}^{(b)} d_{j(V_b)}^{(a)}}{e_G(V_a, V_b)} \right]_{1 \leqslant i \leqslant n/k} = \left[d_{\min} - d_{i(V_a)}^{(b)} \right]_{1 \leqslant i \leqslant n/k} \quad (a \neq b),$$

because $\sum_{j=1}^{n/k} d_{j(V_b)}^{(a)} = e_G(V_a, V_b)$. Therefore, for $c = 1, \ldots, k$ we have

$$Y'\mathbf{1}_{V_c} = \left[(1 - \delta_{v,V_c})(d_{\min} - e_G(v, V_c)) \right]_{v \in V}. \tag{3.4}$$

Further,

$$A\mathbf{1}_{V_c} = (e_G(v, V_c))_{v \in V}, \text{ and } B\mathbf{1}_{V_c} = 0. \tag{3.5}$$

Combining (3.4) and (3.5), we get

$$(A - B + Y')\mathbf{1}_{V_c} = \left[(1 - \delta_{v,V_c}) d_{\min} \right]_{v \in V}. \tag{3.6}$$

Finally, we let $\xi^{(a,b)} = \mathbf{1}_{V_a} - \mathbf{1}_{V_b} \in \mathbb{R}^n$ $(a, b = 1, \ldots, k)$. Then (3.6) yields

$$(A - B + Y')\xi^{(a,b)} = -d_{\min}\xi^{(a,b)} \quad (a \neq b), \; (A - B + Y')\mathbf{1} = (k-1)d_{\min}\mathbf{1}. \tag{3.7}$$

Let $K \subset \mathbb{R}^n$ be the vector space spanned by $\mathbf{1}$ and the vectors $\xi^{(a,b)}$ $(a \neq b)$. Then $\mathbf{1}_{V_1}, \ldots, \mathbf{1}_{V_k} \in K$, and therefore $\mathbf{1}_{V_1}, \ldots, \mathbf{1}_{V_k}$ generate K. Since by (3.7) K is generated by eigenvectors of $A - B + Y'$, any eigenvector η of $A - B + Y'$ with eigenvalue $< -d_{\min}$ is perpendicular to K.

https://doi.org/10.1017/S0963548306007917 Published online by Cambridge University Press

Thus, the following lemma shows that no eigenvector with eigenvalue $< -d_{\min}$ exists w.h.p., and hence concludes the proof that $L' \leqslant Y$ w.h.p.

Lemma 3.3. *Let $G = G_{n,p,k}$. Then w.h.p. we have $d_{\min} = \Omega(np/k)$ and*

$$|\langle (A - B + Y')\eta, \eta\rangle| < d_{\min}$$

for all unit vectors $\eta \perp K$ and all possible choices of v^*, w^*.

We prove Lemma 3.3 in Section 3.3. Now, suppose that indeed $L' \leqslant Y$. Since $Y = Y' + \mathrm{diag}(y')$ and $\mathrm{diag}(Y') = 0$, we have

$$\sum_{i=1}^{n} y_{ii} = \langle y', \mathbf{1}\rangle = 2\#E(G) + nd_{\min}, \tag{3.8}$$

$$\begin{aligned}\sum_{i\neq j} y_{ij} &= \sum_{a\neq b} \langle Y'_{ab}\mathbf{1}, \mathbf{1}\rangle \overset{(3.4)}{=} \sum_{a\neq b} \frac{n}{k} d_{\min} - e_G(V_a, V_b) \\ &= (k-1)nd_{\min} - 2\#E(G).\end{aligned} \tag{3.9}$$

Combining (3.8) and (3.9), we obtain

$$\begin{aligned}\mathrm{SDP}_h(G') \leqslant \mathrm{DSDP}_h(G') &\leqslant \frac{h-1}{2h}\sum_{i=1}^{n} y_{ii} - \frac{1}{2h}\sum_{i\neq j} y_{ij} \\ &= \frac{h-1}{2k}(2\#E(G) + nd_{\min}) - \frac{1}{2h}((k-1)nd_{\min} - 2\#E(G)) \\ &= \#E(G) - \frac{nd_{\min}}{2h}(k-h).\end{aligned} \tag{3.10}$$

As $d_{\min} = \Omega(np/k)$ w.h.p. by Lemma 3.3, we conclude that Lemma 3.2 holds for $G = G_{n,p,k}$.

Finally, let $G = G^*_{n,p,k}$, and let G_0 be the random k-colourable graph contained in G (*i.e.*, $G \in \mathcal{I}(G_0)$). Let G'_0 (resp. G') be obtained from G_0 (resp. from G) by adding an edge $\{v^*, w^*\}$, $v^*, w^* \in V_i$. Since adding a single edge can increase the value of SDP_h by at most 1, w.h.p. we have

$$\mathrm{SDP}_h(G') \leqslant \mathrm{SDP}_h(G'_0) + \#E(G) - \#E(G_0) \overset{(3.10)}{\leqslant} \#E(G) - \Omega\left(\frac{np}{2hk}\right)(k-h),$$

as desired.

3.3. Proof of Lemma 3.3

To prove the lemma, we decompose the adjacency matrix A of $G = G_{n,p,k}$ into blocks $A = (A_{ab})_{a,b=1,\dots k}$ of size $\frac{n}{k} \times \frac{n}{k}$. Then due to our assumption (3.2) for any two vertices $v \in V_a$ and $w \in V_b$, the $(v - (a-1)\frac{n}{k}), (w - (b-1)\frac{n}{k})$-entry of A_{ab} is 1 if $\{v, w\} \in E(G)$ and 0 if $\{v, w\} \notin E(G)$. In particular, $A^T_{ab} = A_{ba}$ and $A_{aa} = 0$. Moreover, the entries of each block A_{ab} with $a \neq b$ are mutually independent random variables that attain the value 1 with probability p and the value 0 with probability $1 - p$.

Lemma 3.4. *If $G = G_{n,p,k}$, then w.h.p. the following statements hold.*

(1) $\bar{d} = \Theta(np/k)$.

https://doi.org/10.1017/S0963548306007917 Published online by Cambridge University Press

(2) *For all unit vectors* $\eta \perp K$ *we have* $|\langle A\eta, \eta\rangle| \leqslant O(\sqrt{\bar{d}k})$.
(3) *For all* $a, b \in \{1, \dots, k\}$ *and all unit vectors* $\mathbf{1} \perp \xi \in \mathbb{R}^{n/k}$ *we have*

$$\left|\|\mathbf{1}\|^{-1} \cdot \langle A_{ab}\xi, \mathbf{1}\rangle\right| \leqslant O(\sqrt{\bar{d}}).$$

Proof. The first statement is an immediate consequence of the definition of $\bar{d}$ and the Chernoff bound (2.4). Moreover, the third statement is an immediate consequence of Lemma 2.3, because $np/k = \Omega(\ln n)$ by (1.2).

To prove the second statement, let $G_a = G_{n/k,p}$ for $a = 1, \dots, k$ be a family of k mutually independent random graphs. Let $A_a^* = A(G_a)$ be the adjacency matrices ($a = 1, \dots, k$). Moreover, let

$$A_* = \begin{pmatrix} A_1^* & & 0 \\ & \ddots & \\ 0 & & A_k^* \end{pmatrix}$$

be the $n \times n$ matrix with the $\frac{n}{k} \times \frac{n}{k}$-blocks $A_1^*, \dots, A_k^*$ on the diagonal and zeros elsewhere. In addition, set $A^* = A + A_*$. Then A^* is distributed as the adjacency matrix $A(G_{n,p})$ of a random graph $G_{n,p}$. Observe that $\eta \perp K$ implies that $\eta \perp \mathbf{1}$. Thus, since $np = \Omega(\ln n)$ by (1.2), Lemma 2.2 entails that there is a constant $\zeta_1 > 0$ such that w.h.p.

$$\forall \eta \perp K,\ \|\eta\| = 1 : |\langle A^*\eta, \eta\rangle| \leqslant \zeta_1\sqrt{\bar{d}k}. \tag{3.11}$$

Furthermore, decomposing $\eta \perp K$ into k subsequent pieces $\eta_1, \dots, \eta_k \in \mathbb{R}^{n/k}$, we obtain

$$\begin{aligned} |\langle A_*\eta, \eta\rangle| &= \left|\sum_{a=1}^{k} \langle A_a^*\eta_a, \eta_a\rangle\right| \leqslant \sum_{a:\eta_a \neq 0} \|\eta_a\|^2 \cdot \left|\left\langle A_a^* \frac{\eta_a}{\|\eta_a\|}, \frac{\eta_a}{\|\eta_a\|}\right\rangle\right| \\ &\leqslant \|\eta\|^2 \cdot \max_{a:\eta_a \neq 0} \left|\left\langle A_a^* \frac{\eta_a}{\|\eta_a\|}, \frac{\eta_a}{\|\eta_a\|}\right\rangle\right|. \end{aligned} \tag{3.12}$$

If $\eta \perp K$, then $\eta_a \perp \mathbf{1}$ for $a = 1, \dots, k$. Therefore, as $A_a^* = A(G_{n/k,p})$ and $np/k = \Omega(\ln n)$, by Lemma 2.2 there is a constant $\zeta_2 > 0$ such that w.h.p.

$$\forall \eta \perp K,\ \|\eta\| = 1 : |\langle A_*\eta, \eta\rangle| \leqslant \max_{a:\eta_a \neq 0} \left|\left\langle A_a^* \frac{\eta_a}{\|\eta_a\|}, \frac{\eta_a}{\|\eta_a\|}\right\rangle\right| \leqslant \zeta_2\sqrt{\bar{d}}. \tag{3.13}$$

Finally, we claim that w.h.p.

$$\forall \eta \perp K,\ \|\eta\| = 1 : |\langle A\eta, \eta\rangle| \leqslant (\zeta_1 + \zeta_2)\sqrt{\bar{d}k}. \tag{3.14}$$

Indeed, suppose that A violates (3.14). Then there is a unit vector $\eta \perp K$ such that $|\langle A\eta, \eta\rangle| > (\zeta_1 + \zeta_2)\sqrt{\bar{d}k}$. Hence, for all A_* that satisfy (3.13) we have $|\langle A^*\eta, \eta\rangle| \geqslant |\langle A\eta, \eta\rangle| - |\langle A_*\eta, \eta\rangle| > \zeta_1\sqrt{\bar{d}k}$, so that A^* violates (3.11). Since (3.11) and (3.13) hold with probability $1 - o(1)$, we conclude that the probability that (3.14) is violated is $o(1)$, as desired. □

Proof of Lemma 3.3. Since $np \geqslant (1 + \varepsilon)k \ln(n)$, the fact that $d_{\min} = \Omega(\bar{d})$ w.h.p. follows from the Chernoff bound (2.5). Furthermore, we claim that

$$|\langle Y'\eta, \eta\rangle| \leqslant O(\sqrt{\bar{d}k}) \quad \text{w.h.p.} \tag{3.15}$$

https://doi.org/10.1017/S0963548306007917 Published online by Cambridge University Press

Indeed, consider the following $\frac{n}{k} \times \frac{n}{k}$ matrices Z_{ab} $(a, b = 1, \ldots, k)$: we let $Z_{aa} = 0$ for all a, and $Z_{ab} = \frac{k}{n} d_{\min}\mathbf{J} - Y'_{ab}$ $(a \neq b)$. Moreover, let $Z = (Z_{ab})_{a,b=1,\ldots,k}$ be the $n \times n$ matrix consisting of the blocks Z_{ab}. Then for all $\eta \perp K$ we have

$$\langle Z\eta, \eta\rangle = -\langle Y'\eta, \eta\rangle, \tag{3.16}$$

because $\eta \perp \mathbf{1}_{V_a}$ for all $a \in \{1, \ldots, k\}$. Thus, it suffices to estimate $|\langle Z\eta, \eta\rangle|$. Let $\xi = (\xi_i)_{1\leqslant i\leqslant n/k}, \eta = (\eta_i)_{1\leqslant i\leqslant n/k} \in \mathbb{R}^{n/k}$ be unit vectors perpendicular to $\mathbf{1}$. Then

$$\begin{aligned} e_G(V_a, V_b)\langle \eta, Z_{ab}\xi\rangle &= \left\langle \eta, \left[\sum_{j=1}^{n/k} d^{(b)}_{i(V_a)} d^{(a)}_{j(V_b)} \xi_j\right]_{1\leqslant i\leqslant n/k}\right\rangle \\ &= \left\langle \eta, \left[d^{(b)}_{i(V_a)} \langle A_{ba}\mathbf{1}, \xi\rangle\right]_{1\leqslant i\leqslant n/k}\right\rangle \\ &= \langle A_{ba}\mathbf{1}, \xi\rangle \sum_{i=1}^{n/k} d^{(b)}_{i(V_a)} \eta_i = \langle A_{ba}\mathbf{1}, \xi\rangle\langle A_{ab}\mathbf{1}, \eta\rangle \\ &= \langle A_{ab}\xi, \mathbf{1}\rangle\langle A_{ba}\eta, \mathbf{1}\rangle. \end{aligned} \tag{3.17}$$

By the third part of Lemma 3.4, w.h.p. we have

$$|\langle A_{ab}\xi, \mathbf{1}\rangle|,\ |\langle A_{ba}\eta, \mathbf{1}\rangle| \leqslant O\big(\sqrt{\bar{d}n/k}\big) \tag{3.18}$$

for all unit vectors $\xi, \eta \perp \mathbf{1}$ and all a, b. Moreover, since $e_G(V_a, V_b)$ is binomially distributed with mean $n^2k^{-2}p$, the Chernoff bound (2.4) and the first part of Lemma 3.4 entail that w.h.p.

$$e_G(V_a, V_b) = \Omega(\bar{d}n/k) \qquad (1 \leqslant a < b \leqslant k). \tag{3.19}$$

Combining (3.17)–(3.19), we get

$$|\langle Z_{ab}\xi, \eta\rangle| \leqslant \frac{O(\bar{d}n/k)}{e_G(V_a, V_b)} = O(1) \quad \text{w.h.p.} \qquad (1 \leqslant a < b \leqslant k). \tag{3.20}$$

Thus, let $\eta \perp K$ be a unit vector. Decomposing η into k pieces $\eta_1, \ldots, \eta_k \in \mathbb{R}^{n/k}$, we get

$$\begin{aligned} |\langle Z\eta, \eta\rangle| &= \left|\sum_{a,b=1}^{k} \langle Z_{ab}\eta_b, \eta_a\rangle\right| \leqslant \sum_{a,b:\eta_a\neq 0\neq \eta_b} \|\eta_a\| \cdot \|\eta_b\| \cdot \left|\left\langle Z_{ab}\frac{\eta_b}{\|\eta_b\|}, \frac{\eta_a}{\|\eta_a\|}\right\rangle\right| \\ &\overset{(3.20)}{\leqslant} O\left[\sum_{a,b=1}^{k} \|\eta_a\| \cdot \|\eta_b\|\right] \leqslant O(1)\left[\sum_{a=1}^{k} \|\eta_a\|\right]^2 = O(k) \\ &\overset{(1.2)}{\leqslant} o\big(\sqrt{\bar{d}k}\big). \end{aligned} \tag{3.21}$$

Combining (3.16) and (3.21), we obtain (3.15).

As $\|B\| \leqslant 2$, the second part of Lemma 3.4 yields in combination with (3.15) that w.h.p.

$$\forall \eta \perp K,\ \|\eta\| = 1 : |\langle (A - B + Y')\eta, \eta\rangle| \leqslant O\big(\sqrt{\bar{d}k}\big). \tag{3.22}$$

As $d_{\min} = \Omega(\bar{d})$ and $\bar{d} = \Theta(np/k)$, (1.2) and (3.22) give

$$\forall \eta \perp K,\ \|\eta\| = 1 : |\langle (A - B + Y')\eta, \eta\rangle| < d_{\min},$$

provided that the constant C_0 is sufficiently large. □

https://doi.org/10.1017/S0963548306007917 Published online by Cambridge University Press

4. Colouring $G^*_{n,p,k}$ in polynomial expected time

In this section we present the algorithm `ExpColour` for Theorem 1.3. After exhibiting a few properties of $G^*_{n,p,k}$ in Section 4.1, we outline the algorithm `ExpColour` and its subroutines in Section 4.2. Sections 4.3–4.5 contain the technical details of the analysis of `ExpColour`.

Throughout, we write $G = G^*_{n,p,k}$. *We assume that* (1.4) *is satisfied with a sufficiently large constant* C_0.

4.1. Basic properties of $G^*_{n,p,k}$

Let $U \subset \{1, \ldots, k\}$, and consider the graph $G[V_U]$ induced on the colour classes V_i with $i \in U$. Let $u = \#U$. Then $G[V_U]$ is u-colourable, so that $\mathrm{SDP}_u(G[V_U]) = \#E(G[V_U])$ (see (2.3)). Now let G' be a graph obtained from G by adding random edges *inside* the colour classes V_i, $i \in U$. The following lemma, which we prove in Section 7.1, shows that these additional random edges do not increase the value of SDP_u 'too much'. More precisely, we have $\mathrm{SDP}_u(G') - \mathrm{SDP}_u(G) = O(\frac{nu}{k}\sqrt{np})$; note that by (1.4) $\#E(G[V_U])$ is binomially distributed with mean $\binom{u}{2}n^2k^{-2}p \gg \frac{nu}{k}\sqrt{np}$.

Lemma 4.1. *Let $U \subset \{1, \ldots, k\}$ be a set of cardinality $u = \#U$. With probability $\geqslant 1 - \exp(-100nu/k)$ the graph $G = G^*_{n,p,k}$ enjoys the following property.*

Let G' be a graph obtained from G by adding each edge inside the colour classes V_i with probability p independently. Then for a certain constant $C_1 > 0$ we have

$$\mathrm{P}\left[\mathrm{SDP}_u(G'[V_U]) \leqslant \#E(G[V_U]) + C_1\frac{nu}{k}\sqrt{np}\right] \geqslant 2/3, \tag{4.1}$$

where probability is taken over the choice of the random edges inside the colour classes.

We will use Lemma 4.1 in Section 4.2 in order to investigate the geometric structure of rigid vector colourings of G.

Now consider a single colour class V_i. The subgraph of $G = G^*_{n,p,k}$ consisting only of the V_i–$V \setminus V_i$-edges contains a random bipartite graph. Hence, we expect that this bipartite graph is a good 'expanding graph'. To quantify the expansion property of this graph precisely, we need the following concept. Let $T \subset V \setminus V_i$, and let $\eta \geqslant 0$. A set M of T–V_i-edges of G is a *d-fold matching with defect* $\leqslant \eta$ from T to V_i if there exists a set $D \subset T$, $\#D \leqslant \eta$, such that:

- every vertex in $T \setminus D$ is incident with precisely d edges in M, and
- every vertex in V_i is incident with at most one edge in M.

Now, we define the *defect* $\mathrm{def}_G(V_i)$ as follows (see [12, Section 2.3]).

D1. If there is a subset $U \subset V_i$ of cardinality $\#U \geqslant \frac{n}{2k}$ such that $\#V \setminus (V_i \cup N_G(U)) > \frac{n}{200k^2}$, then we let $\mathrm{def}_G(V_i) = \frac{n}{2k}$.

D2. Otherwise, we let $\mathrm{def}_G(V_i)$ be the least number $0 \leqslant \eta \leqslant \frac{n}{2k}$ such that for all $6 \leqslant d \leqslant \lceil 50k \rceil$ the following holds: every set $T \subset V \setminus V_i$ of size $\#T \leqslant \frac{n}{2dk}$ admits a d-fold matching to V_i with defect $\leqslant \eta$.

The interpretation of D1 is the following. Let $U \subset V_i$ be a fixed set of cardinality $\#U \geqslant \frac{n}{2k}$. Then, for each $v \in V_j$, $j \neq i$, the probability that v has no neighbour in U is $(1-p)^{\#U} \leqslant \exp(-np/(2k)) \leqslant \exp(-\Omega(k))$. Hence, we *expect* that U is connected to all but at most

https://doi.org/10.1017/S0963548306007917 Published online by Cambridge University Press

$n\exp(-\Omega(k))$ vertices outside of V_i. Thus, D1 just says that we consider G a rather 'bad' expander (*i.e.*, the defect is large) if there is some U such that the non-neighbourhood of U in $V \setminus V_i$ is $\geqslant \frac{n}{200k^2}$, *i.e.*, much bigger than the expected $n\exp(-\Omega(k))$. Moreover, D2 basically says that if the defect is small, then every (reasonably small) set $T \subset V \setminus V_i$ is connected to V_i by a lot of edges. Thus, $\mathrm{def}_G(V_i)$ quantifies the expansion of the bipartite graph consisting of the V_i–$V \setminus V_i$-edges: the smaller the defect is, the better is the expansion.

The following lemma bounds the probability that the defect gets large.

Lemma 4.2. *Let $\eta_i \geqslant 0$ for $i = 1, \ldots, k$. Then*

$$\mathrm{P}(\mathrm{def}_G(V_i) \geqslant \eta_i \textit{ for } i = 1, \ldots, k) \leqslant \prod_{i=1}^{k} \binom{n/k}{\eta_i}^{-100}.$$

We prove Lemma 4.2 in Section 7.2. Furthermore, in Section 7.3 we prove the following lemma, which shows that every sufficiently large independent set of $G^*_{n,p,k}$ consists mainly of vertices from one colour class w.h.p.

Lemma 4.3. *With probability $\geqslant 1 - \exp(-100n)$ the semirandom graph $G = G^*_{n,p,k}$ enjoys the following property.*

$$\begin{aligned} &\textit{If } U \textit{ is an independent set in } G \textit{ of size } \#U \geqslant \tfrac{n}{100k},\\ &\textit{then } \#U \cap V_i > \tfrac{199}{200}\#U \textit{ for some } 1 \leqslant i \leqslant k. \end{aligned} \tag{4.2}$$

Moreover, with probability $\geqslant 1 - \exp(-100n/\ln k)$ the graph $G = G^*_{n,p,k}$ satisfies the following condition for all $i \in \{1, \ldots, k\}$.

$$\textit{If } U \subset V_i,\ \#U \geqslant \tfrac{n}{2k\ln(k)}, \textit{ then } \#\bar{N}_G(U) \leqslant \tfrac{2n}{k}. \tag{4.3}$$

4.2. Outline

In order to k-colour G, `ExpColour`(G, k) (see Figure 2) runs the procedure `Classes`, which proceeds recursively in k stages. In each stage, `Classes` tries to recover one of the colour classes $V_1, \ldots, V_k$, and then hands the graph without the recovered colour class to the next stage. More precisely, if W_l is the set of vertices that have not yet been coloured in the previous stages, then the lth stage tries to exhibit a set $\mathcal{S}_l$ of large independent sets of $G[W_l]$. Then for each $S_l \in \mathcal{S}_l$, `Classes` passes the graph $G[W_l \setminus S_l]$ to stage $l+1$, which tries to find a $(k-l)$-colouring of this graph. If G is 'typical', which happens with high probability, then each $\mathcal{S}_l$ will consist precisely of one colour class, so that a k-colouring will be found immediately.

However, since our goal is an algorithm that k-colours *all* k-colourable graphs, we also have to deal with 'atypical' input instances G. To this end, `ExpColour` uses the variable T, which controls the size of the 'search tree' that `ExpColour` is building, *i.e.*, what amount of running time `ExpColour` spends in order to k-colour G. This amount of time is distributed among the k stages of `Classes` via the variables $\eta_1, \ldots, \eta_k$. The variable η_{k-l+1} determines for how 'typical' the lth stage takes its input graph: the larger η_{k-l+1}, the less 'typical' the graph is assumed to be. In order to (try to) produce a set $\mathcal{S}_l$ that contains one of the hidden colour classes,

https://doi.org/10.1017/S0963548306007917 Published online by Cambridge University Press

---

Algorithm 2. `ExpColour`(G, k)
Input: A graph $G = (V, E)$, an integer $k \geqslant \chi(G)$. *Output:* A k-colouring of G.

1. For $T = 1, \dots, \lfloor \exp(n/\ln k) \rfloor$ do
2. Let $\eta = \max\{\xi \in \mathbb{Z} : \exp(\xi), \binom{n/k}{\xi} \leqslant T\}$.
   For each decomposition $\eta = \eta_1 + \cdots + \eta_k$ where $0 \leqslant \eta_i \leqslant \frac{n}{2k}$ are integers such that $\prod_{i=1}^{k} \binom{n/k}{\eta_i} \leqslant T$ do
3. If `Classes`$(G, V, k, \eta_1, \dots, \eta_k)$ k-colours G, then output the colouring and halt.
4. For $T = \lceil \exp(n/\ln k) \rceil, \dots, \lfloor \exp(n) \rfloor$ do
   If `Exact`(G, k, T) k-colours G, then output the resulting colouring and halt.
5. Colour G optimally via Lawler's algorithm [31] in time $O(2.443^n)$.

Figure 2. The algorithm `ExpColour`.

stage l of `ExpColour` may spend time $\left(n\binom{n/k}{\eta_{k-l+1}}\right)^{O(1)}$. Thus, as the variable T grows from 1 to $\exp(n)$, the running time increases 'smoothly' from polynomial to exponential.

In addition to `Classes`, `ExpColour` has a further subroutine `Exact`. This procedure is used as a fallback if `Classes` does not k-colour G before T exceeds $\exp(n/\ln k)$.

In order to analyse `ExpColour`, we shall assign to each graph $G = G^*_{n,p,k}$ a value T^* such that `ExpColour` k-colours G before T exceeds T^*. Then, on the one hand we can bound the running time of `ExpColour`(G, k) in terms of T^*. On the other hand, we shall investigate the distribution of T^* to prove that the *expected* running time is polynomial.

4.2.1. The procedure `Classes`. The input of `Classes` (see Figure 3) consists of the graph G, a set $W \subset V(G)$, the number k, and integers $\eta_1, \dots, \eta_l$. `Classes` is to find an l-colouring of $G[W]$. In steps 1–3, `Classes` computes a set $\mathcal{S}_l$ of independent sets of $G_l = G[W]$, each of cardinality n/k. Then, in steps 4–5, `Classes` tentatively colours each of the sets $S_l \in \mathcal{S}_l$ with the lth colour, and calls itself recursively on input $(G, W \setminus S_l, k, \eta_1, \dots, \eta_{l-1})$ in an attempt to $(l-1)$-colour $G[W \setminus S_l]$.

Suppose the input graph G is a semirandom graph $G^*_{n,p,k}$ with hidden colouring $V_1, \dots, V_k$. Like the heuristic `Colour` in Section 3, `Classes` employs the relaxation $\bar{\vartheta}_2$ of the chromatic number (see Section 2 for the definition), but in a more sophisticated way. If $\eta_l < \frac{n}{2k}$, then step 2 of `Classes` tries to use the rigid vector colouring $(x_v)_{v \in W}$ to recover a large independent set S_v (see Lemma 4.5 below). By Lemma 4.3, with extremely high probability S_v consists mainly of vertices of one colour class V_i. Then, to recover V_i from S_v, `Classes` uses a further procedure `Purify` (see Corollary 4.6 below).

However, if $\eta_l \geqslant \frac{n}{2k}$, then step 3 of `Classes` assumes that $\bar{\vartheta}_2$ behaves 'badly', so that the aforementioned approach is hopeless. Instead, step 3 enumerates all subsets U of W of cardinality $\frac{n}{k \ln k}$ and considers their non-neighbourhoods. Eventually, step 3 will encounter a set U that lies entirely inside a colour class V_i. By the second part of Lemma 4.3, we expect that $\#\bar{N}_{G_l}(U) \leqslant \frac{2n}{k}$. If so, step 3 adds all independent subsets of $\bar{N}_{G_l}(U)$ of cardinality $\frac{n}{k}$ to $\mathcal{S}_l$. Thus,

https://doi.org/10.1017/S0963548306007917 Published online by Cambridge University Press

Algorithm 3. `Classes`$(G, W, k, \eta_1, \ldots, \eta_l)$
Input: A graph $G = (V, E)$, a set $W \subset V$, integers $k, \eta_1, \ldots, \eta_l$.
Output: Either an l-colouring of $G[W]$ or 'fail'.

1. Let $G_l = G[W]$.
   If $l = 1$ and G_l is an independent set, then return a 1-colouring of G_l.
   If $\bar{\vartheta}_2(G_l) > l$, then return 'fail'.
   Otherwise, compute a rigid vector l-colouring $(x_v)_{v \in W}$ of G_l.
2. If $\eta_l < \frac{n}{2k}$
   If for all $w \in W$ the set $S_w = \{u \in W : \langle x_u, x_w \rangle \geqslant 0.99\}$ has cardinality $< \frac{199n}{200k}$, return 'fail'.
   Otherwise, let $v = \min\{w \in W : \#S_w \geqslant \frac{199n}{200k}\}$.
   Let $\mathcal{S}_l =$ `Purify`$(G, S_v, \eta_l, n/k)$.
3. else
   Let $\mathcal{S}_l = \emptyset$. For each $U \subset W$, $\#U = \frac{n}{2k\ln(k)}$, do
   Let $T = \bar{N}_{G_l}(U)$.
   If $\#T \leqslant 2n/k$, then for all $I \subset T$, $\#I = n/k$, do
   If I is an independent set, then add I to $\mathcal{S}_l$.
4. For each $S_l \in \mathcal{S}_l$ do
5. If `Classes`$(G, W \setminus S_l, k, \eta_1, \ldots, \eta_{l-1})$ $(l-1)$-colours $G[W \setminus S_l]$, return the l-colouring of G_l obtained by colouring S_l with an lth colour.
6. Return 'fail'.

Figure 3. The procedure `Classes`.

as U is contained in the independent set V_i, we have $V_i \subset \bar{N}_{G_l}(U)$, so that the colour class V_i will be added to $\mathcal{S}_l$. The following proposition summarizes the analysis of `Classes`.

Proposition 4.4. *To each semirandom graph $G = G^*_{n,p,k}$ that satisfies properties* (4.2) *and* (4.3) *we can associate a sequence $(\eta^*_1, \ldots, \eta^*_k) \in \left\{0, 1, \ldots, \frac{n}{2k}\right\}^k$ such that the following two conditions hold.*

(1) `Classes`$(G, V, k, \eta^*_1, \ldots, \eta^*_k)$ *outputs a k-colouring of G.*

(2) *Let $\eta_1, \ldots, \eta_k \geqslant 0$. Then $\mathrm{P}(\eta^*_i \geqslant \eta_i$ for all $i) \leqslant \prod_{i=1}^k \binom{n/k}{\eta_i}^{-90}$.*

The running time of `Classes`$(G, V, k, \eta_1, \ldots, \eta_k)$ *is at most $n^{O(1)} \prod_{i=1}^k \binom{n/k}{\eta_i}^{14}$.*

The crucial insight behind `Classes` is that w.h.p. we can use the rigid vector colouring to recover a 'large' independent set of size $\frac{199n}{200k}$ (see step 2). By Lemma 4.3, such an independent set will consist mainly of vertices from one of the planted colour classes, *i.e.*, in the case of success we have recovered a huge fraction of one colour class. In order to extract a large independent set from the vector colouring, the basic idea is as follows. Imagine throwing random edges into the colour classes of $G = G^*_{n,p,k}$ by including the edges inside the colour classes V_i with probability p independently. (Of course, the *algorithm* cannot do this, because it does not know the colour classes yet.) Let G' be the resulting graph. How do $\mathrm{SDP}_k(G)$ and $\mathrm{SDP}_k(G')$ compare? By Lemma 4.1, with probability $\geqslant \frac{2}{3}$ over the choice of the random edges inserted into the colour classes, $\mathrm{SDP}_k(G')$ exceeds $\mathrm{SDP}_k(G) = \#E(G)$ by at most $O(n^{3/2}p^{1/2})$. Hence, if $(x_v)_{v\in V}$ is a rigid vector k-colouring of G, then there are only $O(n^{3/2}p^{1/2})$ random edges $\{v, w\}$ inside the colour classes V_i whose contribution $1 - \langle x_v, x_w\rangle$ to the sum (see (2.3))

$$\mathrm{SDP}_k(G) \leqslant \left(1 - \frac{1}{k}\right) \sum_{\{s,t\}\in E(G')} 1 - \langle x_s, x_t\rangle \leqslant \mathrm{SDP}_k(G')$$

is 'large', say, $1 - \langle x_v, x_w\rangle \geqslant 1/200$. But then we can derive from our assumption (1.4) that there is at least one colour class such that for almost all vertices v, w in this class the vectors x_v, x_w are 'close to each other', say, $\langle x_v, x_w\rangle \geqslant 0.99$. In fact, these vertices can be found easily by 'guessing' one of them, say w, and considering all the vertices that are close to it, *i.e.*, the set S_w. The following lemma makes this idea rigorous.

Lemma 4.5. *Let $G = G^*_{n,p,k}$. Assume that property* (4.1) *holds for the set $U \subset \{1, \dots, k\}$, $\#U = u > 1$. Let $(x_v)_{v\in V_U}$ be a rigid vector u-colouring of $G[V_U]$. Then there is a vertex $v \in V_U$ such that $S_v = \{w \in V_U : \langle x_v, x_w\rangle \geqslant 0.99\}$ is an independent set of cardinality $\geqslant \frac{199n}{200k}$ in G.*

Proof. Consider the graph $H = (V_U, F)$, where $F = \{\{v, w\} : \langle x_v, x_w\rangle < 0.99\}$. Then $G[V_U]$ is a subgraph of H, because $\langle x_v, x_w\rangle < 0$ for all edges $\{v, w\} \in E(G)$. Let $\mathcal{B} = \bigcup_{i\in U} E(H[V_i])$ be the set of all edges of H that join two vertices that belong to the same colour class of G. Let $b = \#\mathcal{B}$.

Furthermore, let G' be the random graph obtained from G by including each V_i–V_i-edge with probability p independently for all $i \in \{1, \dots, k\}$. Note that $(x_v)_{v\in V}$ is a feasible solution to SDP_u. Hence, by property (4.1), with probability $\geqslant 2/3$ over the choice of the random edges inside the colour classes, we have

$$\sum_{\{v,w\}\in E(G'[V_U])} \frac{u-1}{u}(1 - \langle x_v, x_w\rangle) \leqslant \mathrm{SDP}_u(G'[V_U]) \leqslant \#E(G[V_U]) + C_1\frac{nu}{k}\sqrt{np}. \tag{4.4}$$

Observe that an edge $e = \{v, w\}$ of $G'[V_U]$ contributes 1 to the sum on the left-hand side if $e \in E(G)$, and that e contributes $\geqslant \frac{1}{200}$ if $e \in \mathcal{B}$. Therefore, (4.4) entails that

$$\mathrm{P}\left(\#\mathcal{B} \cap E(G'[V_U]) \leqslant 200C_1\frac{nu}{k}\sqrt{np}\right) \geqslant \frac{2}{3}. \tag{4.5}$$

We claim that $b \leqslant \frac{u}{401}n^2k^{-2}$. Indeed, assume for contradiction that $b > \frac{u}{401}n^2k^{-2}$. Then (1.4) yields that $bp > 2000C_1nuk^{-1}\sqrt{np}$, provided that the constant C_0 is large enough. Since $\#\mathcal{B} \cap E(G'[V_U])$ is binomially distributed with mean bp, by the Chernoff bound (2.4) we obtain

$$\mathrm{P}\left(\#\mathcal{B} \cap E(G'[V_U]) > 200C_1\frac{nu}{k}\sqrt{np}\right) \geqslant \mathrm{P}\left(\#\mathcal{B} \cap E(G'[V_U]) \geqslant \frac{bp}{10}\right) \geqslant \frac{1}{2},$$

contradicting (4.5).

https://doi.org/10.1017/S0963548306007917 Published online by Cambridge University Press

Algorithm 4. $\texttt{Exact}(G, k, T)$
Input: A graph $G = (V, E)$, an integer $k \geqslant \chi(G)$, an integer $T \geqslant 0$.
Output: Either a k-colouring of G or 'fail'.

1. Let $0 \leqslant x \leqslant n$ be the largest integer such that $\binom{n}{x}k^x k! \leqslant T$.
   For each triple (X, φ, σ), where
   - $X \subset V$, $\#X = x$,
   - φ is a k-colouring of $G[X]$,
   - σ is a permutation of $\{1, \dots, k\}$

   do the following.
2. For $l = 1, \dots, k$ do
   If $V \setminus (S_1 \cup \cdots \cup S_{l-1})$ has no independent subset of size n/k, then abort the 'for' loop and try the next triple (X, φ, σ).
   Otherwise, let S_l' be the lexicographically first subset of $V \setminus (S_1 \cup \cdots \cup S_{l-1})$ of size n/k that is independent in G. Then, let
   $$S_l = (S_l' \cup \varphi^{-1}(\sigma_l)) \setminus \varphi^{-1}(\{1, \dots, k\} \setminus \{\sigma_l\}).$$
3. If $(S_1, \dots, S_k)$ is a k-colouring of G, then output this colouring and halt.
4. Answer 'fail'.

Figure 4. The procedure $\texttt{Exact}$.

Thus, $b \leqslant \frac{u}{401}n^2k^{-2}$. Consequently, there is some $i \in U$ and a vertex $v \in V_i$ such that v has degree $< \frac{n}{200k}$ in $H[V_i]$. Hence, $S_v = \bar{N}_H(v)$ has size $\#S_v \geqslant \frac{199n}{200k}$. Furthermore, as for all $w, w' \in S_v$ we have $\langle x_v, x_w\rangle, \langle x_v, x_{w'}\rangle \geqslant 0.99$, we obtain that $\langle x_w, x_{w'}\rangle \geqslant 0$. Therefore, $\{w, w'\} \notin E(G)$, so that S_v is an independent set in G. □

In addition to the relaxation $\bar{\vartheta}_2$, step 2 of $\texttt{Classes}$ employs a procedure $\texttt{Purify}$ from [12]. The following corollary is a reformulation of [12, Proposition 2.6] for the present setting.

Corollary 4.6. *Let $G = G^*_{n,p,k}$. Let $i \in \{1, \dots, k\}$. Suppose that I is an independent set that satisfies $\#I \cap V_i \geqslant \frac{99n}{100k}$. Further, assume that $\mathrm{def}_G(V_i) \leqslant \eta < \frac{n}{2k}$. Then the output $\mathcal{S}$ of $\texttt{Purify}(G, I, \eta, n/k)$ contains V_i as an element, and its running time is $\leqslant n^{O(1)}\binom{n/k}{\eta}^{14}$.*

Thus, suppose that step 2 of $\texttt{Classes}$ recovers a large independent set S_v from the rigid vector colouring such that $\#S_v$ already contains 99% of the vertices of some colour class V_i. Then Corollary 4.6 entails that $\texttt{Purify}$ will in fact recover the actual class V_i, provided that the parameter η exceeds the defect $\mathrm{def}_G(V_i)$. Combining Lemma 4.5 and Corollary 4.6, we prove Proposition 4.4 in Section 4.3.

4.2.2. The procedure $\texttt{Exact}$. If $\texttt{Classes}$ fails to k-colour the input graph $G = G^*_{n,p,k}$, then $\texttt{ExpColour}$ calls the procedure $\texttt{Exact}$ (see Figure 4). The goal of $\texttt{Exact}$ is to exhibit a k-colouring of G in expected time $n^{O(1)}\exp(n/\ln k)$. Thus, the expected running time of $\texttt{Exact}$

is somewhat smaller than the worst case running time $\exp(\Theta(n))$ of known exact colouring algorithms. In Section 4.4 we prove the following proposition.

Proposition 4.7. *Let $G = (V, E)$ be a k-colourable graph, and let* $\exp(n/\ln k) \leqslant T \leqslant \exp(n)$.

(1) $\texttt{Exact}(G, k, T)$ *either outputs a k-colouring of G or 'fail'.*
(2) *The running time of* $\texttt{Exact}(G, k, T)$ *is* $\leqslant n^{O(1)}T^4$.
(3) *If $G = G^*_{n,p,k}$, then the probability that* $\texttt{Exact}(G = G^*_{n,p,k}, k, T)$ *answers 'fail' is* $\leqslant T^{-90}$.

The idea behind `Exact` is to 'guess' a certain part of the hidden colouring of $G = G^*_{n,p,k}$. More precisely, `Exact` enumerates all sets X of a suitably chosen size x, and all k-colourings φ of X; the k-colouring $\varphi : X \to \{1, \dots, k\}$ that `Exact` is really interested in is the one induced by the planted colouring of G. Then, in step 2, `Exact` tries to find large independent sets S'_l of G. By property (4.2), we expect that each of these sets consists mainly of vertices in one colour class V_{σ_l}. Using the 'guess' (X, φ, σ), `Exact` tries to correct the set S'_l so that $S_l = V_{\sigma_l}$: step 2 removes all vertices in $\varphi^{-1}(\{1, \dots, k\} \setminus \{\sigma_l\})$, *i.e.*, all vertices from the other classes that have erroneously ended up in S'_l, and adds all vertices in $\varphi^{-1}(\sigma_l)$, *i.e.*, all missing vertices from V_{σ_l}. The size of the 'guess' of `Exact` is ruled by the parameter T. Note that the choice of x ensures that the number of possible triples (X, φ, σ) is $\leqslant T$.

4.3. Proof of Proposition 4.4

Given $G = G^*_{n,p,k}$, we define the sequence $\eta^* = (\eta^*_1, \dots, \eta^*_k)$ of numbers $\eta^*_i \in \{0, \dots, \frac{n}{2k}\}$ along with a permutation σ of $\{1, \dots, k\}$ inductively as follows. Having defined η^*_i and σ_i for $i = 1, \dots, l-1$, we let

$$U_l = \{1, \dots, k\} \setminus \{\sigma_1, \dots, \sigma_{l-1}\}. \tag{4.6}$$

If the graph $G[V_{U_l}]$ does not satisfy property (4.1), then we let $\eta^*_l = n/(2k)$, and let $\sigma_l = \min U_l$. Otherwise, let $(x_v)_{v \in V_{U_l}}$ be the rigid vector $(k-l+1)$-colouring of $G[V_{U_l}]$ computed by step 1 of `Classes` on input G and $W = V_{U_l}$. By Lemma 4.5, there is a vertex w such that $\#S_w \geqslant \frac{199n}{200k}$. Let

$$v_l = \min\left\{ w \in V_{U_l} : \#S_w \geqslant \frac{199n}{200k} \right\} \tag{4.7}$$

be the smallest such vertex (recall that $V_{U_l} \subset V = \{1, \dots, n\}$). As we assume that G has property (4.2), there is a unique $\sigma_l \in \{1, \dots, k\}$ such that $\#V_{\sigma_l} \cap S_{v_l} > \frac{99n}{100k}$. Now, we let

$$\eta^*_l = \mathrm{def}_G(V_{\sigma_l}), \tag{4.8}$$

and proceed inductively.

The following lemma establishes the first part of Proposition 4.4. Throughout, we assume that G has properties (4.2) and (4.3).

Lemma 4.8. $\texttt{Classes}(G, V, k, \eta^*_k, \eta^*_{k-1}, \dots, \eta^*_1)$ *finds a k-colouring of G.*

https://doi.org/10.1017/S0963548306007917 Published online by Cambridge University Press

Proof. We show by induction that eventually $V_{\sigma_l} \in \mathcal{S}_l$ for $l = 1, \ldots, k$. Assume that the algorithm sets $S_j = V_{\sigma_j}$ for $j = 1, \ldots, l-1$. We show that then the set $\mathcal{S}_l$ computed by

$$\texttt{Classes}\left[G, W = V \setminus \bigcup_{j=1}^{l-1} V_{\sigma_j}, k, \eta_k^*, \ldots, \eta_l^*\right]$$

contains V_{σ_l} as an element. There are two cases to consider.

Case 1: $\eta_l^* < n/(2k)$. `Classes` executes step 2. Thus, by (4.7), step 2 picks $v = v_l$. Moreover, due to (4.8) we have $\eta_l^* = \mathrm{def}_G(V_{\sigma_l}) < n/(2k)$. Therefore, Corollary 4.6 entails that the output $\mathcal{S}_l$ of `Purify` contains V_{σ_l} as an element.

Case 2: $\eta_l^* \geqslant n/(2k)$. Eventually, step 3 will encounter some $U \subset V_{\sigma_l}$, $\#U = \frac{n}{2k \ln k}$. By property (4.3), we have $\#\bar{N}_{G_l}(U) \leqslant 2n/k$. As $V_{\sigma_l} \subset \bar{N}_{G_l}(U)$, step 3 adds V_{σ_l} to $\mathcal{S}_l$.

Thus, in both cases we have $V_{\sigma_l} \in \mathcal{S}_l$, so that eventually step 4 will try $S_l = V_{\sigma_l}$. Then step 5 calls

$$\texttt{Classes}\left[G, V \setminus \bigcup_{j=1}^{l} V_{\sigma_j}, k, \eta_k^*, \ldots, \eta_{l+1}^*\right].$$

Hence, proceeding inductively, we conclude that the colouring $(V_{\sigma_1}, \ldots, V_{\sigma_k})$ will be recovered. □

Moreover, the second assertion in Proposition 4.4 follows from the next lemma.

Lemma 4.9. *Let $\eta_1, \ldots, \eta_k \geqslant 0$ be integers. Let $\eta_1^*, \ldots, \eta_k^*$ be as defined above, with input graph $G = G^*_{n,p,k}$. Then $\mathrm{P}[\eta_i^* \geqslant \eta_i$ for $i = 1, \ldots, k] \leqslant \prod_{i=1}^k \binom{n/k}{\eta_i}^{-90}$.*

Thus, Lemma 4.9 estimates the probability that the random variables $\eta_1^*, \ldots, \eta_k^*$ exceed certain values $\eta_1, \ldots, \eta_k$.

Proof of Lemma 4.9. Fix a sequence $\eta_1, \ldots, \eta_k \geqslant 0$ of integers. Let $G = G^*_{n,p,k}$ be a semirandom graph with planted colouring $V_1, \ldots, V_k$. Given integers $0 \leqslant l, \lambda \leqslant k$, we define an event $\mathcal{E}(l, \lambda)$ as follows: $G \in \mathcal{E}(l, \lambda)$ if and only if there exist two disjoint sets $J_1, J_2 \subset \{1, \ldots, k\}$ and an injective map $\tau : J_1 \cup J_2 \to \{1, \ldots, k\}$ such that the following conditions are satisfied.

E1. $\#J_1 = l$ and $\#J_2 = \lambda$.
E2. For $U = V_{\tau(J_1)}$ property (4.1) is violated.
E3. $\mathrm{def}_G(V_{\tau_i}) \geqslant \eta_i \geqslant 1$ for all $i \in J_2$.
E4. $\eta_i = 0$ for all $i \in \{1, \ldots, k\} \setminus (J_1 \cup J_2)$.

We shall prove below that

$$\mathrm{P}[\mathcal{E}(l, \lambda)] \leqslant \prod_{i=1}^{k} \binom{n/k}{\eta_i}^{-98} \quad \text{for all } l, \lambda. \tag{4.9}$$

Furthermore, we claim that if $G = G^*_{n,p,k}$ is such that $\eta_i^* \geqslant \eta_i$ for $i = 1, \ldots, k$, then there exist l^*, λ^* so that $G \in \mathcal{E}(l^*, \lambda^*)$. For if property (4.1) does not hold in G with $U = \{1, \ldots, k\}$,

https://doi.org/10.1017/S0963548306007917 Published online by Cambridge University Press

then $G \in \mathcal{E}(k, 0)$. Otherwise, we can define

$$l^* = k - \max\left\{1 \leqslant l \leqslant k : \text{property (4.1) holds in } G[V_{U_j}] \text{ for all } j \leqslant l\right\}, \tag{4.10}$$

where U_l is defined in (4.6). In addition, we set

$$J_1 = \{k - l^* + 1, \dots, k\} = \sigma^{-1}(U_{k-l^*+1}), \; J_2 = \{1 \leqslant i \leqslant k - l^* : \eta_i > 0\},$$

and we define $\tau : J_1 \cup J_2 \to \{1, \dots, k\}$, $j \mapsto \sigma_j$. Then by (4.8) and (4.10), J_1, J_2, and τ satisfy E1–E4 with respect to $\mathcal{E}(l^*, \lambda^* = \#J_2)$. Hence, $G \in \mathcal{E}(l^*, \lambda^*)$. As a consequence, if $\max_{i=1,\dots,k} \eta_i > 0$, then we obtain

$$\begin{aligned} \mathrm{P}[\eta_i^* \geqslant \eta_i \text{ for } i = 1, \dots, k] &\leqslant \sum_{l,\lambda=1}^{k} \mathrm{P}[\mathcal{E}(l, \lambda)] \overset{(4.9)}{\leqslant} k^2 \cdot \prod_{i=1}^{k} \binom{n/k}{\eta_i}^{-98} \\ &\overset{(1.4)}{\leqslant} \prod_{i=1}^{k} \binom{n/k}{\eta_i}^{-90}, \end{aligned}$$

as desired.

Thus, the remaining task is to prove (4.9). If we fix sets J_1, J_2, and an injection $\tau : J_1 \cup J_2 \to \{1, \dots, k\}$ such that E1 and E4 hold, then by Lemma 4.1 and Lemma 4.2 we have

$$\begin{aligned} \mathrm{P}[\text{E2, E3 occur}] &\leqslant \exp\left(-\frac{100ln}{k}\right) \prod_{i \in J_2} \binom{n/k}{\eta_i}^{-100} \\ &\leqslant \exp\left(-\frac{ln}{k}\right) \prod_{i=1}^{k} \binom{n/k}{\eta_i}^{-99}. \end{aligned} \tag{4.11}$$

Further, there are

$$\leqslant \binom{k}{l}\binom{k}{\lambda} k^l \lambda! \leqslant k^{2l+\lambda} \tag{4.12}$$

ways to choose J_1, J_2, and τ subject to E1 and E4. (For there are $\leqslant \binom{k}{l}$ ways to choose J_1 and $\leqslant \binom{k}{\lambda}$ ways to choose J_2. Given J_1 and J_2, there are $\leqslant k^l$ ways to choose the restriction of τ to J_1, and finally $\leqslant \lambda!$ choices of the restriction of τ to J_2 subject to E4.) Combining (4.11) and (4.12), we obtain

$$\begin{aligned} \mathrm{P}[\mathcal{E}(l, \lambda)] &\leqslant k^{2l+\lambda} \cdot \exp\left(-\frac{ln}{k}\right) \prod_{i=1}^{k} \binom{n/k}{\eta_i}^{-99} \leqslant k^{\lambda} \cdot \left(\frac{k}{n}\right)^{\lambda} \cdot \prod_{i=1}^{k} \binom{n/k}{\eta_i}^{-98} \\ &\leqslant \left(\frac{k^2}{n}\right)^{\lambda} \cdot \prod_{i=1}^{k} \binom{n/k}{\eta_i}^{-98} \overset{(1.4)}{\leqslant} \prod_{i=1}^{k} \binom{n/k}{\eta_i}^{-98}, \end{aligned}$$

thereby establishing (4.9). □

Proof of Proposition 4.4. Since the first two assertions follow from Lemmas 4.8 and 4.9, we just need to bound the running time of $\texttt{Classes}(G, V, k, \eta_1, \dots, \eta_k)$. Clearly, step 1 runs in polynomial time. Moreover, by Corollary 4.6, the total time spent on executing step 2 (for all k

https://doi.org/10.1017/S0963548306007917 Published online by Cambridge University Press

stages) is

$$\leqslant n^{O(1)} \prod_{i=1}^{k} \binom{n/k}{\eta_i}^{14}. \tag{4.13}$$

Further, step 3 consumes time

$$n^{O(1)} \binom{n}{n/(2k \ln k)} \binom{2n/k}{n/k} \leqslant n^{O(1)} \binom{n/k}{n/(2k)}^{7}; \tag{4.14}$$

for there are $\leqslant \binom{n}{n/(2k\ln k)}$ ways to choose the set U, and if $\#T \leqslant 2n/k$, then there are $\leqslant \binom{2n/k}{n/k}$ ways to choose the set I. Since step 3 gets executed only if $\eta_l = \frac{n}{2k}$, (4.13) and (4.14) entail that the total running time is

$$\leqslant n^{O(1)} \left[\prod_{i=1}^{k} \binom{n/k}{\eta_i}^{14} + \prod_{i=1}^{k} \binom{n/k}{\eta_i}^{7} \right] \leqslant n^{O(1)} \prod_{i=1}^{k} \binom{n/k}{\eta_i}^{14},$$

so that the proposition follows. □

4.4. Proof of Proposition 4.7

Due to steps 3–4, $\texttt{Exact}(G, k, T)$ either outputs a k-colouring of its input graph G or 'fail'. Thus, the remaining task is to bound the running time of $\texttt{Exact}$ on input $G^*_{n,p,k}$.

Thus, let $G = G^*_{n,p,k}$, and let $V_1, \ldots, V_k$ be the planted k-colouring of G. Assuming property (4.2), we construct

- sets $X^*_l \subset V$ of cardinality $2x^*_l$ for $l = 1, \ldots, k$,
- a permutation σ^* of $\{1, \ldots, k\}$,
- independent sets $S'^*_1, \ldots, S'^*_k$

inductively as follows. Starting with $l = 1$, let $S'^*_l \subset V \setminus \bigcup_{j=1}^{l-1} V_{\sigma^*_l}$ be the lexicographically first independent set of cardinality n/k. Then by property (4.2), there is a $1 \leqslant i \leqslant k$ such that $\#V_i \cap S'^*_l > n/(2k)$. Set $\sigma^*_l = i$, let

$$X^*_l = (V_i \setminus S'^*_l) \cup (S'^*_l \setminus V_i), \tag{4.15}$$

$$x^*_l = \#S'^*_l \setminus V_i, \tag{4.16}$$

and proceed inductively. Finally, set $X^* = \bigcup_{l=1}^{k} X^*_l$ and $x^* = \#X^*$. Then

$$x^* \leqslant 2 \sum_{i=1}^{k} x^*_i. \tag{4.17}$$

Further, let φ^* be the colouring induced on X^* by the k-colouring $(V_1, \ldots, V_k)$ of G, and set

$$T^* = \binom{n}{x^*} k^{x^*} k!.$$

If property (4.2) is violated in G, then we let $T^* = \lceil \exp(n) \rceil$.

Lemma 4.10. *If $T^* < \lceil \exp(n) \rceil$, then $\texttt{Exact}(G, k, T)$ outputs a k-colouring for all $T \geqslant T^*$.*

https://doi.org/10.1017/S0963548306007917 Published online by Cambridge University Press

Proof. We claim that at the latest when $T = T^*$, $(X, \varphi, \sigma) = (X^*, \varphi^*, \sigma^*)$, steps 2–3 of `Exact` will k-colour G. The proof is by induction on $l = 1, \ldots, k$. Suppose that `Exact` has set $S_j = V_{\sigma_j}$ for all $1 \leqslant j < l$. To show that then $S_l = V_{\sigma_l}$, we let $S'_l \subset V \setminus \bigcup_{j=1}^{l-1} S_j = V \setminus \bigcup_{j=1}^{l-1} V_{\sigma_j}$ be the independent set of size n/k computed in step 2. As S'_l is the lexicographically first independent set of size n/k, by construction we have $S'_l = S'^*_l$. Hence, (4.15) entails that

$$(V_{\sigma_l} \setminus S'_l) \cup (S'_l \setminus V_{\sigma_l}) = X^*_l \subset X^* = X. \tag{4.18}$$

As $\sigma = \sigma^*$ and $\varphi = \varphi^*$ is the k-colouring induced by $(V_1, \ldots, V_k)$ on X^*, we get

$$\begin{aligned} S_l &= (S'_l \cup \varphi^{-1}(\sigma_l)) \setminus \varphi^{-1}(\{1, \ldots, k\} \setminus \{\sigma_l\}) \\ &= (S'_l \cup (X^* \cap V_{\sigma_l})) \setminus (X^* \setminus V_{\sigma_l}) \overset{(4.18)}{=} V_{\sigma_l}, \end{aligned}$$

as desired. Thus, steps 2–3 find a k-colouring. □

Lemma 4.11. *Let* $\exp(n/\ln k) \leqslant T \leqslant \lceil \exp(n) \rceil$. *Then* $\mathrm{P}[T^* > T] \leqslant T^{-90}$.

Proof. If G violates property (4.2), then $T^* = \lceil \exp(n) \rceil$, and the assertion follows from Lemma 4.3. Thus, we may assume that property (4.2) holds. Set

$$x^{(T)} = \max\left\{x \geqslant 0 : \forall\, 0 \leqslant y \leqslant x : T \geqslant \binom{n}{y} k^y k!\right\},$$

$$\mathcal{X}^{(T)} = \left\{(x_1, \ldots, x_k) \in \{0, \ldots, x^{(T)}\}^k : \sum_{i=1}^{k} x_i = \left\lceil \frac{x^{(T)}}{2} \right\rceil\right\}.$$

Since (1.4) entails

$$\binom{n}{100k/p} k^{100k/p} k! \leqslant \binom{n}{100k/p}^2 k^{200k/p} \leqslant \left(\frac{enp}{100}\right)^{200k/p} \leqslant \exp\left(\frac{n}{\ln k}\right) \leqslant T,$$

we conclude that

$$x^{(T)} \geqslant \frac{100k}{p}. \tag{4.19}$$

Given a sequence $x_1, \ldots, x_k$ of integers $\geqslant 0$, we consider the following event $\mathcal{E}(x_1, \ldots, x_k)$:

> There is a permutation σ of $\{1, \ldots, k\}$ and a collection of sets $S''_1, \ldots, S''_k$ such that $S''_l \subset V_{\sigma_l}$, $\#S''_l \geqslant \frac{n}{2k}$, and $\#\bar{N}_G(S''_l) \setminus V_{\sigma_l} \geqslant x_i$ for $l = 1, \ldots, k$.

We shall prove below that for all $(x_1, \ldots, x_k) \in \mathcal{X}^{(T)}$

$$\mathrm{P}[\mathcal{E}(x_1, \ldots, x_k)] \leqslant \exp\left(2n - \left[\frac{np}{4k} - \ln n\right] \frac{x^{(T)}}{2}\right). \tag{4.20}$$

Now, let $G = G^*_{n,p,k}$ be such that $T^* > T$. Let $S''_l = S'^*_l \cap V_{\sigma^*_l}$. Because the sets S'^*_l are independent, we have $S'^*_l \setminus V_{\sigma^*_l} \subset \bar{N}_G(S''_l)$ $(l = 1, \ldots, k)$. Moreover, by construction we have $\#S''_l \geqslant \frac{n}{2k}$, so that $\mathcal{E}(x^*_1, \ldots, x^*_k)$ occurs (see (4.16)). Further, as $T^* > T$ and $x^* \leqslant n/2$, we have

$$x^{(T)} < x^* \overset{(4.17)}{\leqslant} 2 \sum_{i=1}^{k} x^*_i.$$

Reducing some of the x_i^*s if necessary, we obtain a sequence $(x_1, \ldots, x_k) \in \mathcal{X}^{(T)}$ such that $\mathcal{E}(x_1, \ldots, x_k)$ occurs. Thus,

$$\begin{aligned} \mathrm{P}[T^* > T] &\leqslant \sum_{(x_1,\ldots,x_k)\in\mathcal{X}^{(T)}} \mathrm{P}[\mathcal{E}(x_1,\ldots,x_k)] \\ &\overset{(4.20)}{\leqslant} \#\mathcal{X}^{(T)} \cdot \exp\left(2n - \left[\frac{np}{4k} - \ln n\right]\frac{x^{(T)}}{2}\right). \end{aligned} \tag{4.21}$$

Observe that

$$\#\mathcal{X}^{(T)} \leqslant \binom{x^{(T)} + k - 1}{k - 1} \leqslant 2^{x^{(T)}+k-1}, \tag{4.22}$$

$$T \leqslant \binom{n}{x^{(T)} + 1} k^{x^{(T)}+1} k! \overset{(4.19)}{\leqslant} \exp(4x^{(T)} \ln n). \tag{4.23}$$

Therefore, continuing (4.21), we get

$$\begin{aligned} \mathrm{P}[T^* > T] &\overset{(4.22)}{\leqslant} 2^{x^{(T)}+k-1} \exp\left(2n - \left[\frac{np}{4k} - \ln n\right]\frac{x^{(T)}}{2}\right) \\ &\overset{(4.19)}{\leqslant} \exp\left(2x^{(T)}\ln(n) + 2n - \frac{np}{8k}x^{(T)}\right) \overset{(1.4),(4.19)}{\leqslant} \exp\left(-\frac{np}{16k}x^{(T)}\right) \\ &\overset{(1.4),(4.23)}{\leqslant} T^{-90}, \end{aligned}$$

as desired.

Finally, let us prove (4.20). Let $(x_1, \ldots, x_k) \in \mathcal{X}^{(T)}$. Let us fix sets $S_l' \subset V_{\sigma_l}$ of cardinality $\geqslant \frac{n}{2k}$ for a moment. In addition, consider sets $X_1, \ldots, X_k$ such that $X_l \subset V \setminus V_{\sigma_l}$ and $\#X_l = x_l$ for $l = 1, \ldots, k$. If $e_G(X_l, S_l') = 0$ for $l = 1, \ldots, k$, then we know of $\geqslant \frac{1}{2}\sum_{l=1}^k \#X_l \cdot \#S_l' \geqslant \frac{nx^{(T)}}{4k}$ edges that are not present in $G = G^*_{n,p,k}$, although each of these edges occurs with probability p independently in $G_{n,p,k}$. Hence,

$$\mathrm{P}[e_G(X_l, S_l') = 0 \text{ for } l = 1, \ldots, k] \leqslant (1-p)^{nx^{(T)}/(4k)} \leqslant \exp\left(-\frac{x^{(T)}np}{4k}\right). \tag{4.24}$$

Furthermore, given the permutation σ, there are at most $2^{n/k}$ ways to choose the set S_l'. Moreover, there are at most $\binom{n}{x_l}$ ways to choose the set X_l. Therefore, the union bound and (4.24) yield

$$\begin{aligned} \mathrm{P}[\mathcal{E}(x_1,\ldots,x_k)] &\leqslant \exp\left(-\frac{x^{(T)}np}{4k}\right) \cdot k! \prod_{l=1}^k \binom{n}{x_l} 2^{n/k} \\ &\leqslant k! \exp\left(n + \left[\ln(n) - \frac{np}{4k}\right]x^{(T)}\right) \\ &\overset{(1.4)}{\leqslant} \exp\left(2n + \left[\ln(n) - \frac{np}{4k}\right]x^{(T)}\right), \end{aligned}$$

so that (4.20) follows. □

https://doi.org/10.1017/S0963548306007917 Published online by Cambridge University Press

Proof of Proposition 4.7. The first assertion is immediate. For $\exp(n/\ln k) \leqslant T \leqslant \lceil \exp(n) \rceil$, steps 1–3 of $\texttt{Exact}(G, k, T)$ consume time

$$\leqslant n^{O(1)}\left[\binom{n}{x}k^x k!\right] \cdot \binom{n}{n/k} \leqslant n^{O(1)}T \cdot \exp\left(\frac{2n}{\ln k}\right) \leqslant n^{O(1)}T^3;$$

for there are $\leqslant \binom{n}{x}k^x k!$ ways to choose the triple (X, φ, σ), and step 2 needs to check $\leqslant \binom{n}{n/k}$ subsets of V. Hence, the proposition follows from Lemma 4.10 and Lemma 4.11. □

4.5. Proof of Theorem 1.3

By Proposition 4.7, $\texttt{ExpColour}$ computes a k-colouring of every k-colourable input graph. Thus, the remaining task is to show that $\texttt{ExpColour}(G^*_{n,p,k}, k)$ runs in polynomial expected time.

Given $1 \leqslant T \leqslant \exp(n/\ln k)$, we can bound the running time of steps 2–3 of $\texttt{ExpColour}$ as follows. There are at most $z_\eta = \binom{\eta+k-1}{k-1}$ ways to choose the numbers $\eta_1, \ldots, \eta_k$. If $\eta \geqslant k-1$, then

$$z_\eta \leqslant 2^{\eta+k-1} \leqslant \exp(2\eta) \leqslant T^2, \tag{4.25}$$

by the definition of η. Moreover, if $\eta < k-1$, then due to (1.4) we have

$$z_\eta = \binom{\eta+k-1}{\eta} \leqslant \left(\frac{2ek}{\eta}\right)^\eta \leqslant \left(\frac{n}{k\eta}\right)^{2\eta} \leqslant \binom{n/k}{\eta}^2 \leqslant T^2. \tag{4.26}$$

Further, having fixed $(\eta_1, \ldots, \eta_k)$, by Proposition 4.4 step 3 consumes time

$$\leqslant n^{O(1)} \prod_{i=1}^{k} \binom{n/k}{\eta_i}^{14} \leqslant n^{O(1)}T^{14}.$$

Thus, the total running time of steps 2–3 for a given T is

$$R_T \leqslant z_\eta \cdot n^{O(1)}T^{14} \overset{(4.25),(4.26)}{\leqslant} n^{O(1)}T^{16}. \tag{4.27}$$

Now let $G = G^*_{n,p,k}$. Then we define a number $T^* = T^*(G^*_{n,p,k})$ as follows. If G violates property (4.2), then we set $T^*_1 = \exp(n)$. Otherwise, if G violates property (4.3), then we let $T^*_1 = \exp(n/\ln k)$. Moreover, if G satisfies both property (4.2) and (4.3), then let $(\eta^*_1, \ldots, \eta^*_k)$ be as in Proposition 4.4 and set $T^*_1 = \prod_{i=1}^k \binom{n/k}{\eta^*_i}$. In addition, let

$$T^*_2 = \min\left\{\exp(n/\ln k) \leqslant T \leqslant \lceil \exp(n) \rceil : \texttt{Exact}(G, k, T) \text{ finds a } k\text{-colouring of } G\right\}.$$

Set $T^* = T^*_1$, if $T^*_1 \leqslant \exp(n/\ln k)$, and $T^* = \max\{T^*_1, T^*_2\}$ otherwise. Then by Proposition 4.4 and Proposition 4.7, $\texttt{ExpColour}$ finds a k-colouring of G before the variable T exceeds T^*.

Combining Propositions 4.4 and 4.7 and Lemmas 4.1 and 4.3, we conclude that

$$\mathrm{P}[T^* > T] \leqslant T^{-80} \tag{4.28}$$

for all $1 \leqslant T \leqslant \lceil \exp(n) \rceil$. Consequently, by (4.27) and (4.28) we get

$$\sum_{T=1}^{\lfloor \exp(n/\ln k) \rfloor} R_T \mathrm{P}[T^* > T] \leqslant n^{O(1)} \sum_{T=1}^{\infty} T^{16-80} = n^{O(1)},$$

https://doi.org/10.1017/S0963548306007917 Published online by Cambridge University Press

so that the expected time spent on executing steps 1–3 of `ExpColour` is polynomial. Finally, if $T^* \geqslant \exp(n/\ln k)$, then by Proposition 4.7 the expected time spent on executing steps 4–5 of `ExpColour` is polynomial as well. Thus, `ExpColour`$(G^*_{n,p,k}, k)$ runs in polynomial expected time.

5. Colouring $G^*_{n,p,k}$ optimally in polynomial expected time

The goal of this section is to prove Theorem 1.4. *Throughout, we assume that* (1.5) *holds.* The algorithm `OptColour` is based on the following observation.

Lemma 5.1. *We have* $\mathrm{P}\big[\bar\vartheta_2(G^*_{n,p,k}) \leqslant k - \frac{1}{2}\big] \leqslant \exp(-100n)$.

Proof. Let $G_0 = G_{n,p,k}$, and let $G = G^*_{n,p,k} \in \mathcal{I}(G_0)$ be a semirandom graph obtained from G_0. Let $h = k - \frac{1}{2}$. If $\bar\vartheta_2(G_0) \leqslant \bar\vartheta_2(G) \leqslant h$, then G_0 has a rigid vector h-colouring $(x_v)_{v\in V}$. Plugging the feasible solution $(x_v)_{v\in V}$ into the semidefinite program SDP_h, we conclude that

$$\text{if } \bar\vartheta_2(G_0) \leqslant h, \text{ then } \mathrm{SDP}_h(G_0) \geqslant \#E(G_0) \qquad \text{(see (2.3))}. \tag{5.1}$$

Furthermore, as $\#E(G_0)$ is binomially distributed with mean $(1-k^{-1})n^2p/2$, (2.4) entails that

$$\mathrm{P}\left[\#E(G_0) \leqslant (1-k^{-1})\frac{n^2p}{2} - Cn^{3/2}p^{1/2}\right] \leqslant \exp(-101n), \tag{5.2}$$

where $C > 0$ denotes a suitable constant. Combining (5.1) and (5.2), we conclude that

$$\begin{aligned}
\mathrm{P}[\bar\vartheta_2(G^*_{n,p,k}) \leqslant h] &\leqslant \exp(-101n) + \mathrm{P}\left[\mathrm{SDP}_h(G_{n,p,k}) \geqslant (1-k^{-1})\frac{n^2p}{2} - Cn^{3/2}p^{1/2}\right] \\
&\leqslant \exp(-101n) + \mathrm{P}\left[\mathrm{SDP}_h(G_{n,p,k}) \geqslant (1-h^{-1})\frac{n^2p}{2} + \frac{n^2p}{4k^2} - Cn^{3/2}p^{1/2}\right] \\
&\overset{(1.5)}{\leqslant} \exp(-101n) + \mathrm{P}\left[\mathrm{SDP}_h(G_{n,p,k}) \geqslant (1-h^{-1})\frac{n^2p}{2} + \left(\frac{\sqrt{C_0}}{4} - C\right)n^{3/2}p^{1/2}\right].
\end{aligned} \tag{5.3}$$

Choosing the constant C_0 large enough, we can ensure that $\sqrt{C_0}/4 - C \geqslant \sqrt{C_0}/8$. Furthermore, as the random graph $G_{n,p}$ can be obtained from $G_{n,p,k}$ by adding random edges inside the planted colour classes $V_1, \ldots, V_k$, the monotonicity property (2.2) entails that $\mathrm{SDP}_h(G_{n,p,k})$ is stochastically dominated by $\mathrm{SDP}_h(G_{n,p})$. Hence, (5.3) yields

$$\mathrm{P}\big[\bar\vartheta_2(G^*_{n,p,k}) \leqslant h\big] \leqslant \exp(-101n) + \mathrm{P}\left[\mathrm{SDP}_h(G_{n,p}) \geqslant (1-h^{-1})\frac{n^2p}{2} + \frac{\sqrt{C_0}}{8}n^{3/2}p^{1/2}\right]. \tag{5.4}$$

Finally, Lemma 2.1 entails that

$$\mathrm{P}\left[\mathrm{SDP}_h(G_{n,p}) \geqslant (1-h^{-1})\frac{n^2p}{2} + \frac{\sqrt{C_0}}{4}n^{3/2}p^{1/2}\right] \leqslant \exp(-101n), \tag{5.5}$$

provided that the constant C_0 is sufficiently large. Thus, the assertion follows from (5.4) and (5.5). □

Algorithm 5. `OptColour`(G)
Input: A graph $G = (V, E)$. *Output:* An optimal colouring of G.

1. Let $\kappa = \lceil \bar{\vartheta}_2(G) \rceil$.
2. Call `ExpColour`(G, κ) and output the resulting colouring.

Figure 5. The algorithm `OptColour`.

Given an input graph $G = (V, E)$, the algorithm `OptColour` just computes the lower bound $\kappa = \lceil \bar{\vartheta}(G) \rceil$ on $\chi(G)$ and calls `ExpColour`(G, κ) (see Figure 5).

Proof of Theorem 1.4. If $\kappa = k$, then Theorem 1.3 shows that `ExpColour`(G, κ) finds a κ-colouring of $G = G^*_{n,p,k}$. Furthermore, if $\kappa < k$, then either steps 1–4 of `ExpColour`(G, κ) will colour G with κ colours, or step 5 of `ExpColour` computes an optimal colouring of G. Hence, as $\kappa \leqslant \chi(G)$, in any case `OptColour` outputs an optimal colouring. Finally, the fact that the expected running time of `OptColour` is polynomial follows from Theorem 1.3 and Lemma 5.1. $\square$

6. Proof of Theorem 1.2

Consider a random graph $G_0 = G_{n,p,k}$, and let $V_1, \ldots, V_k$ be its colour classes, where $\#V_i = n/k$. Let $p = dk/n$, $d = (\frac{1}{2} - \varepsilon)\ln(n/k)$. Let $i, j_1, j_2 \in \{1, \ldots, k\}$ be distinct. Then for each $v \in V_i$, the number $d_v^{(j_1)}$ of neighbours of v in V_{j_1} has binomial distribution with parameters n/k and p. Therefore,

$$\mathrm{P}(d_v^{(j_1)} = 0) = \mathrm{P}(d_v^{(j_2)} = 0) = (1-p)^{n/k} \sim \exp(-d) = \left(\frac{n}{k}\right)^{\varepsilon - 1/2}.$$

Thus, $\mathrm{P}(d_v^{(j_1)} = d_v^{(j_2)} = 0) \sim (n/k)^{2\varepsilon-1}$. Hence, the expected number of vertices $v \in V_i$ satisfying $d_v^{(j_1)} = d_v^{(j_2)} = 0$ is $\sim (n/k)^{2\varepsilon}$. Since edges are chosen independently, by the Chernoff bound (2.4) the number of such vertices is in fact $\geqslant (\frac{n}{k})^{\varepsilon}$ w.h.p. Consequently, w.h.p. there are sets $S_i \subset V_i$, $\#S_i = \left(\frac{n}{k}\right)^{\varepsilon}$, $i = 1, 2, 3$, such that $N_{G_0}(S_i) \cap (V_1 \cup V_2 \cup V_3) = \emptyset$.

Now assume that we had an algorithm $\mathcal{A}$ that can k-colour $G^*_{n,p,k}$ w.h.p. Let H be an arbitrary graph that admits a 3-colouring with colour classes of cardinality $\left(\frac{n}{k}\right)^{\varepsilon}$ each. We show how to convert $\mathcal{A}$ into a randomized algorithm that 3-colours H, which is NP-hard. First, randomly partition $V = \{1, \ldots, n\}$ into k sets $V_1, \ldots, V_k$ of cardinality n/k. Then choose $\left(\frac{n}{k}\right)^{\varepsilon}$ vertices S_i from V_i at random for $i = 1, 2, 3$. Further, form a complete k-partite graph on the vertices $V_1 \setminus S_1, V_2 \setminus S_2, V_3 \setminus S_3, V_4, \ldots, V_k$, and connect $S_1 \cup S_2 \cup S_3$ completely with $V_4 \cup \cdots \cup V_k$. Finally, embed a randomly permuted copy of H into the set $S_1 \cup S_2 \cup S_3$ (without taking care of the colouring of H, of course). Let G be the resulting graph. We claim that running $\mathcal{A}(G)$ yields a k-colouring of G w.h.p.

https://doi.org/10.1017/S0963548306007917 Published online by Cambridge University Press

To prove the claim, we volunteer as an adversary that given $G_0 = G_{n,p,k}$ produces the instance G described above. (To this end, we may use unlimited computational power.) Given a random graph $G_0 = G_{n,p,k}$ with colour classes $V_1, \ldots, V_k$, we first look for sets $S_i \subset V_i$ such that $\#S_i = \left(\frac{n}{k}\right)^{\varepsilon}$ and $N(S_i) \cap (V_1 \cup V_2 \cup V_3) = \emptyset$, $i = 1, 2, 3$. As pointed out above, such sets S_1, S_2, S_3 exist w.h.p.; if not, we give up. Then, we turn $G_0 - (S_1 \cup S_2 \cup S_3)$ into a complete k-partite graph, and connect $S_1 \cup S_2 \cup S_3$ completely with $V_4 \cup \cdots \cup V_k$. Further, we compute a 3-colouring of the worst-case instance H with colour classes of equal size, permute the vertices in each of the three colour classes of H randomly, and map the three colour classes onto S_1, S_2, S_3 (thus, this time we respect the colouring). The *distribution* of the resulting graph G' coincides with the distribution of the graph G constructed in the previous paragraph, so that $\mathcal{A}$ k-colours G w.h.p. As any k-colouring of G induces a 3-colouring on $G[S_1 \cup S_2 \cup S_3] = H$, we have shown that a polynomial time algorithm for k-colouring $G^*_{n,p,k}$ w.h.p. yields a randomized algorithm for 3-colouring the worst-case instance H.

Remark. The only difference between the above construction and the ones given in [16] is that instead of reducing the problem of k-colouring a k-colourable graph to k-colouring $G^*_{n,p,k}$, we reduced the problem of 3-colouring a 3-colourable graph to k-colouring $G^*_{n,p,k}$. The idea of working a worst-case instance into the semirandom instance occurs already in [4].

7. Proofs of auxiliary lemmas

Throughout, we assume that (1.4) holds for some large enough constant $C_0 > 0$.

7.1. Proof of Lemma 4.1

By Lemma 2.1 there exists a constant $C_1 > 0$ such that

$$\mathrm{P}\left[\mathrm{SDP}_u(G_{un/k,p}) \leqslant \left(1 - \frac{1}{u}\right)\binom{un/k}{2}p + \frac{C_1}{2}\frac{un}{k}\sqrt{np}\right] \geqslant 1 - \exp\left[-\frac{201un}{k}\right]. \tag{7.1}$$

Now consider $G_0 = G_{n,p,k}$, and let G_0' be a graph obtained from G_0 by adding each edge inside the planted colour classes V_i, $i = 1, \ldots, k$, with probability p independently. Then G_0' is distributed as a random graph $G_{n,p}$. Thus, in particular,

$$G_0'[V_U] = G_{un/k,p}. \tag{7.2}$$

Let $\mathcal{B}$ be the event that $\#E(G_0[V_U]) \geqslant \left(1 - \frac{1}{u}\right)\binom{un/k}{2}p - \frac{C_1 nu}{2k}\sqrt{np}$. Furthermore, let $\mathcal{A}$ be the event that (4.1) is violated for V_U in G_0. Then invoking (7.1) and (7.2), we obtain

$$\begin{aligned}\frac{1}{3}\mathrm{P}(\mathcal{A} \cap \mathcal{B}) &\leqslant \mathrm{P}\left[\mathrm{SDP}_u(G_{un/k,p}) > \left(1 - \frac{1}{u}\right)\binom{un/k}{2}p + \frac{C_1}{2}\frac{un}{k}\sqrt{np}\right] \\ &\leqslant \exp\left(-\frac{201un}{k}\right).\end{aligned} \tag{7.3}$$

Moreover, $\#E(G_0[V_U])$ is binomially distributed with mean

$$\mathrm{E}[\#E(G_0[V_U])] = \binom{u}{2}\left(\frac{n}{k}\right)^2 p \sim \left(1 - \frac{1}{u}\right) \cdot \binom{un/k}{2}p.$$

https://doi.org/10.1017/S0963548306007917 Published online by Cambridge University Press

Hence, choosing the constant $C_1 > 0$ sufficiently large and applying (2.4), we get

$$\mathrm{P}(\mathcal{B}) \geqslant 1 - \exp\left(-\frac{201nu}{k}\right). \tag{7.4}$$

Now, (7.4) implies that $\mathrm{P}(\mathcal{A} \setminus \mathcal{B}) \leqslant \mathrm{P}(\neg\mathcal{B}) \leqslant \exp(-201nu/k)$, which in combination with (7.3) entails

$$\mathrm{P}(\mathcal{A}) = \mathrm{P}(\mathcal{A} \cap \mathcal{B}) + \mathrm{P}(\mathcal{A} \setminus \mathcal{B}) \leqslant 4\exp\left(-\frac{201nu}{k}\right) \leqslant \exp\left(-\frac{200nu}{k}\right). \tag{7.5}$$

Finally, let $G = G^*_{n,p,k}$. Let $G_0 = G_{n,p,k}$ be the random k-colourable graph contained in G. Let $M = \#E(G[V_U]) - \#E(G_0[V_U])$ be the number of edges added by the adversary. Then $\mathrm{SDP}_u(G') \leqslant \mathrm{SDP}_u(G'_0) + M$, because adding one edge can increase the value of SDP_u by at most 1. Therefore, if $\mathrm{SDP}_u(G'_0) \leqslant \#E(G'_0[V_U]) + C_1 \frac{nu}{k}\sqrt{np}$, then

$$\mathrm{SDP}_u(G') \leqslant M + \#E(G'_0[V_U]) + C_1 \frac{nu}{k}\sqrt{np} = \#E(G'[V_U]) + C_1 \frac{nu}{k}\sqrt{np}.$$

Hence, the assertion follows from (7.5).

7.2. Proof of Lemma 4.2

Throughout, we fix a partition $(V_1, \ldots, V_k)$ of $V = \{1, \ldots, n\}$ into k disjoint sets of cardinality n/k. Let $G_0 = G_{n,p,k}$ be a random k-colourable graph with planted colouring $V_1, \ldots, V_k$, and let $G = G^*_{n,p,k}$ be the semirandom graph obtained from G_0 by the adversary. Then $\mathrm{def}_G(V_i) \leqslant \mathrm{def}_{G_0}(V_i)$ for $i = 1, \ldots, k$. Hence, it suffices to show that Lemma 4.2 holds for $G = G_0 = G_{n,p,k}$.

Since on $G = G_{n,p,k}$ the random variables $(\mathrm{def}_G(V_i))_{i=1,\ldots,k}$ are not mutually independent, we decompose G into k mutually independent subgraphs $G^{(i)}$ and investigate the defects $\mathrm{def}_{G^{(i)}}(V_i)$ $(i = 1, \ldots, k)$. Letting

$$p' = 1 - \sqrt{1-p} \geqslant p/2, \tag{7.6}$$

we obtain the graph $G^{(i)}$ by including each of the $(1-k^{-1})k^{-1}n^2$ possible $(V \setminus V_i)$–V_i-edges with probability p' independently. Thus, $G^{(i)}$ is a random bipartite graph. Furthermore, in the union

$$H = \bigcup_{i=1}^{k} G^{(i)} = \left(V, \bigcup_{i=1}^{k} E(G^{(i)})\right)$$

every V_i–V_j-edge is present with probability $2p' - p'^2 = p$ independently of all other edges $(i \neq j)$. Therefore, H has the same distribution as $G = G_{n,p,k}$. As a consequence, given $\eta_1, \ldots, \eta_k \geqslant 0$, we have

$$\begin{aligned}
\mathrm{P}[\mathrm{def}_G(V_i) \geqslant \eta_i \text{ for } i = 1, \ldots, k] &= \mathrm{P}[\mathrm{def}_H(V_i) \geqslant \eta_i \text{ for } i = 1, \ldots, k] \\
&\leqslant \mathrm{P}[\mathrm{def}_{G^{(i)}}(V_i) \geqslant \eta_i \text{ for } i = 1, \ldots, k] \\
&= \prod_{i=1}^{k} \mathrm{P}[\mathrm{def}_{G^{(i)}}(V_i) \geqslant \eta_i],
\end{aligned}$$

because the graphs $G^{(i)}$ are mutually independent. Hence, our aim is to prove that

$$\mathrm{P}[\mathrm{def}_{G^{(i)}}(V_i) \geqslant \eta_i] \leqslant \binom{n/k}{\eta_i}^{-100} \quad \text{for } i = 1, \ldots, k. \tag{7.7}$$

To prove (7.7), we first bound the probability that condition D1 in the definition of the defect occurs.

Lemma 7.1. *With probability $\geqslant 1 - \exp(-100n/k)$ the random graph $G^{(i)}$ has the following property.*

$$\text{If } U \subset V_i \text{ has cardinality} \geqslant \tfrac{n}{2k}, \text{ then } \#V \setminus (V_i \cup N_{G^{(i)}}(U)) \leqslant \tfrac{n}{200k^2}. \tag{7.8}$$

Proof. Assuming that $np \geqslant C_0k^2$ for a sufficiently large constant C_0 (see (1.4)), we have $s = 1000/p \leqslant n/(200k^2)$. Fix a set $U \subset V_i$ of cardinality $\geqslant \frac{n}{2k}$ for a moment. Then

$$\begin{aligned} \mathrm{P}[V \setminus (V_i \cup N_{G^{(i)}}(U)) \geqslant s] &\leqslant \binom{n}{s}(1-p')^{ns/(2k)} \overset{(7.6)}{\leqslant} \left(\frac{\mathrm{e}n}{s}\right)^s \exp\left(-\frac{nps}{4k}\right) \\ &\overset{(1.4)}{\leqslant} \exp\left(-\frac{199n}{k}\right). \end{aligned} \tag{7.9}$$

As there are $\leqslant 2^{n/k}$ sets $U \subset V_i$ of cardinality $\geqslant \frac{n}{2k}$, due to the union bound (7.9) entails that

$$\mathrm{P}[(7.8) \text{ is violated}] \leqslant 2^{n/k} \exp\left(-\frac{199n}{k}\right) \leqslant \exp\left(-\frac{198n}{k}\right),$$

and thus the assertion follows. □

Furthermore, the next lemma regards condition D2 in the definition of the defect.

Lemma 7.2. *Let $1 \leqslant \eta \leqslant \frac{n}{2k}$. Then with probability $\geqslant 1 - \binom{n/k}{\eta}^{-101}$ the graph $G^{(i)}$ has the following property.*

Let $6 \leqslant d \leqslant \lceil 50k \rceil$. Then every subset $T \subset V \setminus V_i$ of size $\#T \leqslant \frac{n}{2kd}$ has a d-fold matching to V_i with defect $\leqslant \eta$.

To prove Lemma 7.2, we need the following observation.

Lemma 7.3. *The probability that in $G^{(i)}$ there are $\eta \geqslant 1$ vertices in $V \setminus V_i$ that have $< np'/(2k)$ neighbours in V_i is $\leqslant n^{-200\eta}$.*

Proof. Let $v \in V \setminus V_i$. As $e_{G^{(i)}}(v, V_i)$ is binomially distributed with mean np/k, the Chernoff bound (2.4) yields

$$\mathrm{P}\left[e_{G^{(i)}}(v, V_i) \leqslant \frac{np'}{2k}\right] \leqslant \exp\left(-\frac{np'}{8k}\right).$$

https://doi.org/10.1017/S0963548306007917 Published online by Cambridge University Press

Therefore, the probability that there are $\geqslant \eta \geqslant 1$ such vertices is

$$\leqslant \binom{n}{\eta} \exp\left(-\frac{np'\eta}{8k}\right) \overset{(7.6)}{\leqslant} \exp\left[\left(\ln(n) - \frac{np}{16k}\right)\eta\right] \overset{(1.4)}{\leqslant} n^{-200\eta},$$

as claimed. □

In addition, we need the following lemma from [12, Lemma 4.3].

Lemma 7.4. *Let $V' \subset V$ be a subset of cardinality n_1, and let $V'' = V \setminus V'$, $\#V'' = n_2$. Let $\gamma > 0$ be an arbitrary constant, and let $2 \leqslant d \leqslant n_2/10$. Then there exists a number $\omega_0 = \omega_0(\gamma)$ such that the following holds. Let $\omega = \omega_0 \max\{d, \ln n\}$, and let H be a random bipartite graph obtained as follows: every vertex in V' chooses a set of at least $\omega = \omega_0 \max\{d, \ln n\}$ neighbours in V'' uniformly at random; these choices occur independently for all vertices in V'. Then, for all $\eta \in \{0, 1, \ldots, n_2/2\}$, we have*

$$\mathrm{P}\left[\exists T \subset V' : \#T \leqslant \frac{n_2}{2d} \wedge \#N_H(T) < d\#T - \eta\right] \leqslant \binom{n_2}{\eta}^{-\gamma}.$$

Proof of Lemma 7.2. Fix $6 \leqslant d \leqslant \lceil 50k \rceil$. We say that $G^{(i)}$ is (d, η)-*good* if every set $S \subset V \setminus V_i$ admits a d-fold matching to V_i with defect $\leqslant \eta$. Let $\omega = np'/(2k)$, $W = \{v \in V \setminus V_i : e_{G^{(i)}}(v, V_i) \leqslant \omega\}$, and $G' = G^{(i)}[V \setminus W]$. Let $\omega_0 = \omega_0(110)$ be the number from Lemma 7.4. If we choose the constant C_0 large enough, then (1.4) and (7.6) yield that $\omega \geqslant \omega_0 \cdot \max\{d, \ln n\}$.

Letting $0 \leqslant \eta_1 \leqslant \eta$, we have

$$\mathrm{P}(\#W = \eta_1) \leqslant n^{-200\eta_1} \tag{7.10}$$

by Lemma 7.3. In addition, set $\eta_2 = \eta - \eta_1$. Let us call a set $T \subset V \setminus (V_i \cup W)$ (d, η_2)-*bad* if $\#T \leqslant \frac{n}{2kd}$ and $e_{G^{(i)}}(T, V_i) < d\#T - \eta_2$. Then Lemma 7.4 entails that

$$\mathrm{P}(G^{(i)} \text{ has a } (d, \eta_2)\text{-bad set} | \#W = \eta_1) \leqslant \binom{n/k}{\eta_2}^{-110}. \tag{7.11}$$

If $G^{(i)}$ has no (d, η_2)-bad set, then $G^{(i)}$ is (d, η)-good. For if $S \subset V \setminus V_i$ has size $\#S \leqslant \frac{n}{2dk}$, then Hall's theorem entails that $T = S \setminus W$ has a d-fold matching to T with defect $\leqslant \eta_2$. As a consequence, (7.10) and (7.11) yield

$$\begin{aligned} \mathrm{P}[G^{(i)} \text{ is not } (d, \eta)\text{-good}] \leqslant \sum_{\eta_1=0}^{\eta} & \mathrm{P}[\#W = \eta_1] \\ & \times \mathrm{P}[G^{(i)} \text{ has a } (d, \eta_2)\text{-bad set} | \#W = \eta_1] \\ \leqslant \sum_{\eta_1=0}^{\eta} & n^{-200\eta_1} \binom{n/k}{\eta_2}^{-110} \leqslant \binom{n/k}{\eta}^{-109}. \end{aligned} \tag{7.12}$$

Summing (7.12) over $6 \leqslant d \leqslant \lceil 50k \rceil$, we obtain

$$\mathrm{P}[G^{(i)} \text{ is } (d,\eta)\text{-good for all } 6 \leqslant d \leqslant \lceil 50k \rceil] \geqslant 1 - 50k \binom{n/k}{\eta}^{-109} \overset{(1.4)}{\geqslant} 1 - \binom{n/k}{\eta}^{-108},$$

thereby proving the lemma. □

Combining Lemma 7.1 and Lemma 7.2, we conclude that (7.7) holds, thereby completing the proof of Lemma 4.2.

7.3. Proof of Lemma 4.3

To establish the first part of Lemma 4.3, it suffices to consider random k-colourable graphs $G = G_{n,p,k}$. Let $V_1, \ldots, V_k$ be the planted k-colouring. Let $U \subset V$ be any set of cardinality $u = \#U \geqslant \frac{n}{100k}$ such that

$$u_i = \#U \cap V_i \leqslant \frac{199}{200} u \text{ for } i = 1, \ldots, k. \tag{7.13}$$

Our goal is to bound the probability that U is independent in G. Clearly, $u = \sum_{i=1}^{k} u_i$, and the number of possible edges among the vertices of U is $\lambda = \sum_{1 \leqslant i < j \leqslant k} u_i u_j = \frac{1}{2}\left[u^2 - \sum_{i=1}^{k} u_i^2\right]$. Note that λ is minimized subject to (7.13) when $\sum_i u_i^2$ is maximized. Thus, λ attains its minimal value for $u_1 = 199u/200$, $u_2 = u/200$, and $u_i = 0$ for $i > 2$. Consequently, there is a constant $C_1 > 0$ such that

$$\lambda \geqslant \frac{199}{40000} u^2 \geqslant C_1 \left(\frac{n}{k}\right)^2. \tag{7.14}$$

Hence,

$$\mathrm{P}[U \text{ is independent}] \leqslant (1-p)^{\lambda} \overset{(7.14)}{\leqslant} \exp\left(-C_1 \frac{n^2 p}{k^2}\right) \overset{(1.4)}{\leqslant} \exp(-101n), \tag{7.15}$$

provided that the constant C_0 is large enough. As there are $\leqslant 2^n$ ways to choose U, the assertion follows from the union bound and (7.15).

As for the second assertion, let $G = G_{n,p,k}$ be a random k-colourable graph with planted colouring $V_1, \ldots, V_k$. Consider a set $U \subset V_i$, $\#U \geqslant \frac{n}{2k \ln k}$. Then

$$\mathrm{P}\left[\#\bar{N}_G(U) \setminus V_i > \frac{n}{k}\right] \leqslant \sum_{T \subset V \setminus V_i, \#T = n/k} \mathrm{P}[e_G(U,T) = 0] \leqslant \binom{n}{n/k} (1-p)^{n^2/(2k^2 \ln k)}$$
$$\leqslant \exp\left(\frac{2n}{k} \ln(k) + \frac{n}{k} - \frac{n^2 p}{2k^2 \ln k}\right) \overset{(1.4)}{\leqslant} \exp\left(-\frac{100n}{\ln k}\right),$$

provided that the constant C_0 is sufficiently large.

References

[1] Achlioptas, D. and Naor, A. (2004) The two possible values of the chromatic number of a random graph. In *Proc.* 36th *STOC*, pp. 587–593.

[2] Alon, N. and Kahale, N. (1997) A spectral technique for coloring random 3-colorable graphs. *SIAM J. Comput.* **26** 1733–1748.

[3] Alon, N. and Kahale, N. (1998) Approximating the independence number via the ϑ-function. *Math. Progr.* **80** 253–264.

[4] Blum, A. and Spencer, J. (1995) Coloring random and semirandom k-colorable graphs. *J. Algorithms* **19** 203–234.

[5] Bollobás, B. (1988) The chromatic number of random graphs. *Combinatorica* **8** 49–55.

[6] Bollobás, B. (2001) *Random Graphs*, 2nd edn, Cambridge University Press.

[7] Bollobás, B. and Scott, A. D. (2004) Max cut for random graphs with a planted partition. *Combin. Probab. Comput.* **13** 451–474.

[8] Boppana, R. (1987) Eigenvalues and graph bisection: an average-case analysis. In *Proc.* 28th *FOCS*, pp. 280–285.

[9] Böttcher, J. (2005) Coloring sparse random k-colorable graphs in polynomial expected time. In *Proc.* 30th *Symposium on Mathematical Foundations of Computer Science*, pp. 156–167.

[10] Charikar, M. (2002) On semidefinite programming relaxations for graph coloring and vertex cover. In *Proc.* 13th *SODA*, pp. 616–620.

[11] Coja-Oghlan, A. Solving NP-hard semirandom graph problems in polynomial expected time. *J. Algorithms*, to appear.

[12] Coja-Oghlan, A. (2006) Finding large independent sets in polynomial expected time. *Combin. Probab. Comput.* **15** XXX–XXX.

[13] Coja-Oghlan, A., Moore, C. and Sanwalani, V. (2006) MAX k-CUT and approximating the chromatic number of random graphs. *RSA* **28** 289–322.

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

[15] Feige, U. and Kilian, J. (1998) Zero knowledge and the chromatic number. *J. Comput. System Sci.* **57** 187–199.

[16] Feige, U. and Kilian, J. (2001) Heuristics for semirandom graph problems. *J. Comput. System Sci.* **63** 639–671.

[17] Feige, U. and Krauthgamer, R. (2000) Finding and certifying a large hidden clique in a semirandom graph. *Random Struct. Alg.* **16** 195–208.

[18] Feige, U. and Ofek, E. (2005) Spectral techniques applied to sparse random graphs. *Random Struct. Alg.* **27** 251–275.

[19] Frieze, A. and Jerrum, M. (1997) Improved approximation algorithms for MAX k-CUT and MAX BISECTION. *Algorithmica* **18** 61–77.

[20] Frieze, A. and McDiarmid, C. (1997) Algorithmic theory of random graphs. *Random Struct. Alg.* **10** 5–42.

[21] Goemans, M. X. and Kleinberg, J. (1998) The Lovász theta function and a semidefinite programming relaxation of vertex cover. *SIAM J. Disc. Math.* **11** 1–48.

[22] Goemans, M. X. and Williamson, D. P. (1995) Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. Assoc. Comput. Mach.* **42** 1115–1145.

[23] Grötschel, M., Lovász, L. and Schrijver, A. (1988) *Geometric Algorithms and Combinatorial Optimization*, Springer.

[24] Helmberg, C. (2000) Semidefinite programming for combinatorial optimization. Habilitationsschrift; Report ZR-00-34, Zuse Institute Berlin.

[25] Janson, S., Łuczak, T. and Ruciński, A. (2000) *Random Graphs*, Wiley.

[26] Karger, D., Motwani, R. and Sudan, M. (1998) Approximate graph coloring by semidefinite programming. *J. Assoc. Comput. Mach.* **45** 246–265.

[27] Khanna, S., Linial, N. and Safra, S. (2000) On the hardness of approximating the chromatic number. *Combinatorica* **20** 393–415.

[28] Krivelevich, M. (2002) Coloring random graphs: An algorithmic perspective. In *Proc.* 2nd *Colloquium on Mathematics and Computer Science*, pp. 175–195.

[29] Krivelevich, M. and Vilenchik, D. (2006) Semirandom models as benchmarks for coloring algorithms. In *Proc.* 8th *ANALCO*, pp. 211–221.

[30] Kučera, L. (1989) Graphs with small chromatic number are easy to color. *Inform Process. Lett.* **30** 233–236.

[31] Lawler, E. L. (1976) A note on the complexity of the chromatic number problem. *Inform Process. Lett.* **5** 66–67.

[32] Łuczak, T. (1991) The chromatic number of random graphs. *Combinatorica* **11** 45–54.

[33] McSherry, F. (2001) Spectral partitioning of random graphs. In *Proc.* 42nd *FOCS*, pp. 529–537.

[34] Subramanian, C. R. (1999) Minimum coloring random and semirandom graphs in polynomial average time. *J. Algorithms* **33** 112–123.

[35] Subramanian, C. R., Fürer, M. and Veni Madhavan, C. E (1998) Algorithms for coloring semi-random graphs. *Random Struct. Alg.* **13** 125–158.

[36] Subramanian, C. R. and Veni Madhavan, C. E. (2002) General partitioning on random graphs. *J. Algorithms* **42** 153–172.

[37] Szegedy, M. (1994) A note on the θ number of Lovász and the generalized Delsarte bound. In *Proc.* 35th *FOCS*, pp. 36–39.