MAX-PLANCK-INSTITUT FÜR INFORMATIK

Expected complexity of graph partitioning problems

Luděk Kučera

MPI–I–93–107 February 1993

Im Stadtwald
W 6600 Saarbrücken
Germany

Expected complexity of graph partitioning problems

Luděk Kučera

MPI-I-93-107 February 1993

Expected complexity of graph partitioning problems

Luděk Kučera *
Max Planck Institut für Informatik
Saarbrücken, Germany

February 4, 1993

Abstract

We study the expected time complexity of two graph partitioning problems: the graph coloring and the cut into equal parts.

If $k = o(\sqrt{n/\log n})$, we can test whether two vertices of a k-colorable graph can be k-colored by the same color in time $O(k \log n)$ per pair of vertices with $O(k^4 \log^3 n)$-time preprocessing in such a way that for almost all k-colorable graphs the answer is correct for all pairs of vertices. As a consequence, we obtain a sublinear (with respect to the number of edges) expected time algorithm for k-coloring of k-colorable graphs (assuming the uniform input distribution).

Similarly, if $c \leq (1/8 - \epsilon)n^2$, $\epsilon > 0$ a constant, and G is a graph having cut of the vertex set into two equal parts with at most c cross-edges, we can test whether two vertices belong to the same class of some c-cut in time $O(\log n)$ per vertex with $O(\log^3 n)$-time preprocessing in such a way that for almost all graphs having a c-cut the answer is correct for all pairs of vertices.

The methods presented in the paper can also be used to other graph partitioning problems, e.g. the largest clique or independent subset.

1 Introduction

The present paper investigates the average complexity of two graph theoretic problems: finding the optimal graph coloring and the smallest cut into equal parts. Both problems are well known to be NP-complete, moreover it is known that no good approximation algorithms exist for these problems unless P=NP, and therefore the only way to cope with the problems is to use polynomial time heuristics, and to hope that the result is sufficiently often sufficiently close to the optimal solution.

The expected complexity of any computational problem depends not only on the problem, but on the probabilistic distribution of inputs as well. The simplest input distribution is the uniform one: all inputs of a given size are equally likely. However, this distribution is often uninteresting from the point of view of the algorithm analysis: in most cases the complexity of most inputs is so small, that even very primitive algorithms give very good results. E.g. the expected chromatic number of a graph with n vertices is $(1 + o(1))(n/2\log_2 n)$ ([1, 8]), the greedy coloring algorithm gives

On leave from Charles University, Prague, Czechoslovakia

almost surely a coloring by $(1+o(1))(n/\log_2 n)$ colors [4], and no polynomial time coloring algorithm is known to give better results.

It was shown that it is much more interesting to investigate input distribution that prefer graphs with solution of unlikely size. E.g. a series of papers [5], [9], [6], [7] deals with the uniform distribution on the class of k-colorable graphs with n vertices; interesting results are obtaines for small values k, when the optimal coloring uses much smaller number of colors than it is usual for a general randomly chosen graph with n vertices. It has been proved for such distributions that simple algorithms behave badly, and are not able to find a good solution sufficiently often. E.g. it is shown in [6] that for any fixed $\varepsilon > 0$ the greedy algorithm uses almost surely $(1+o(1))n/\log_2 n$ colors when applied to the uniform distribution on the class of n^ε-colorable graphs with n vertices.

On the other hand, there are more sophisticated algorithms, that are able to find the optimal solution almost surely, provided the bound to the size of solutions is sufficiently different from the expected size of the solution in the class of all graphs with given number of vertices. It was shown that if k is fixed [3], smaller that $(1-\varepsilon)\log_2 n$ [9] or $o(\sqrt{n\log n})$ [7], there are polynomial time algorithms that give almost surely the optimal coloring of a graph drawn uniformly at random from the class of k-colorable graphs with n vertices. The problem of the best cut into equal parts can be solved almost surely optimally by a polynomial time algorithm in the class of graphs with n vertices, n even, and the smallest cut of the size less than $(1/8-\varepsilon)n^2$, $\varepsilon > 0$, see [3]. Let us remind that it is easy to prove that almost all graphs with n vertices have a cut into equal parts of the size $(1/8+\varepsilon)n^2$, but no such cut of size $(1/8-\varepsilon)n^2$.

The complexity of such algorithms was studied by Dyer and Frieze [3]. The aim of the present paper is to improve their results with respect to the time complexity. Both problems studied in the paper try to find a partition of the vertex set (in the case of coloring the partition is given by color classes, and a cut itself is a partition into two sets). Instead of looking for the optimal partition, the primitive question studied in the paper is the following one:

Given two vertices x and y, is there an optimal partition $\mathcal{P}$ such that both x and y belong to the same class (different classes, resp.) of $\mathcal{P}$?

For the problems mentioned above we will present an algorithm that answers the question correctly for any pair of vertices of almost all k-colorable graphs, graphs with the cut of size c, resp. (provided k or c are sufficiently small), and has very low time complexity. In the extreme case (k constant for the coloring problem, $c \le (1/8-\varepsilon)n^2$) for the cut problem, we need $\log^{O(1)} n$ time preprocessing, and $O(\log n)$ time per query.

The rest of the paper is divided into 5 parts. The paragraph 2 introduces input graph distributions for our problems, and gives necessary definitions. The paragraph 3 explains the main idea of the algorithm, which is presented in full in the paragraph 4. The basic algorithm is given as a randomized one, and the paragraph 5 shows how to transform it into a deterministic procedure. The last part of the paper deals with possible applications of our methods to other graph partition problems and distributions.

The main theorems of the paper are Theorem 5.5 and Theorem 5.6. A special case of first result is

Theorem 1.1 *Let $k = o(\sqrt{n/\log n})$. There is a deterministic sequential algorithm that answers queries*

"Given a k-colorable graph G and its vertices x, y, do x, y receive the same color for some k-coloring of G?",
needs $O(k^4 \log^3 n)$ preprocessing time, $O(k \log n)$ time per query, and for almost all k-colorable graphs the answer to all possible queries are correct.

The next statement is essentially a special case of Theorem 5.6, and it could be proved by modification of our methods. However, it deals with slightly different probability distribution, which is easier to be defined, but less convenient input distribution, and therefore its rigorous proof is omitted.

Theorem 1.2 *Let n be even, $c \leq (1/8 - \varepsilon)n^2$, where $0 < \varepsilon$ is a constant, denote by $\mathcal{R}_{n,c}$ the class of all graphs with n vertices, that have a cut of vertices into two equal parts with at most c cross-edges. There is a deterministic sequential algorithm that answers queries*

"Given $G \in \mathcal{R}_{n,c}$ and its vertices x, y, do x, y belong the same class of some c-cut?, needs $O(\log^3 n)$ preprocessing time, $O(\log n)$ time per query, and for almost all graphs from $\mathcal{R}_{n,c}$ the answer to all possible queries are correct.

2 Definitions and lemmas

Definition 2.1 *Given a graph G, its vertices x and y, and a set of vertices Y, we denote the number of neighbours of x in G by $\deg(x)$, the number of neighbours of x in the set Y by $\deg(x, Y)$, and the number of common neighbours of both x and y by $\deg(x, y)$.*

Random graphs that will serve as inputs for algorithms investigated in the paper are defined in a general way for both problems mentioned in the introduction.

We say that an element $\mathbf{x}$ is chosen uniformly from a set X if $\mathbf{Prob}(\mathbf{x} = x) = |X|^{-1}$ for any $x \in X$.

Definition 2.2 *Suppose that we are given two natural numbers n, m, $1 < m \leq n$, real numbers $0 \leq p_{i,j} \leq 1$, $1 \leq i, j \leq m$, such that $p_{i,j} = p_{j,i}$ for all i, j, and natural numbers $n_1, \ldots, n_m$ such that $n_1 + \cdots + n_m = n$.*

Construct a random graph $\mathcal{G}$ as follows: choose uniformly an element $(V_1, \ldots, V_m)$ of the set of all partitions of the set $V = \{1, \ldots, n\}$ into m classes, and connect two vertices $v \in V_i$ and $w \in V_j$ by an edge with probability $p_{i,j}$.

Construct a random graph $\mathcal{H}$ similarly as $\mathcal{G}$, with the difference that the partition is chosen uniformly from the class of all partitions $(V_1, \ldots, V_m)$ such that $|V_i| = n_i$ for $i = 1, \ldots, m$.

Given two vertices $x \in V_i$, $y \in V_j$, denote

$$\Delta(x, y) = \sum_{k=1}^{m} |V_k| p_{i,k} p_{j,k}.$$

When we speak later about distributions $\mathcal{G}$ and $\mathcal{H}$, the sets $V, V_1, \ldots, V_m$ will always have the meaning given in this definition.

Note that the graph $\mathcal{G}$ does not depend on numbers n_i. In the rest of the paper we will suppose that $0 < p < 1$ is a constant (though the results can be modified for p depending on n).

As an input to the coloring problem, we will use random graphs $\mathcal{C}_{n,p,k}$, that are identical with $\mathcal{G}$ for $m = k$, $p_{i,j} = p$ for $i \neq j$, $p_{i,i} = 0$. We will always suppose that $k = o(\sqrt{n/\log n})$. The partition from the definition of $\mathcal{G}$ gives always a coloring of the graph by k colors. It can be proved for $k = o(\sqrt{n/\log n})$ [7] that the partition from the definition is almost surely the unique optimal coloring of $\mathcal{C}_{n,p,k}$, and the probability distribution given by $\mathcal{C}_{n,p,k}$ approximates the uniform distribution on the set of all k-colorable graphs with n-vertices. It will be clear from the proofs given in the paper

that the less natural distribution obtained from the same set of parameters by the construction $\mathcal{H}$ has essentially the same properties, provided $c_1 n/k \leq n_1, \ldots, n_m \leq c_2 n/k$ for some constants $0 < c_1 \leq c_2$.

The problem of the smallest cut into two parts of equal size uses random graphs $\mathcal{R}_{n,p,q}$, where n is even, $q < p < 1$, and $(p-q)^{-1} = o(\sqrt[4]{n/\log n})$, defined by the distribution $\mathcal{H}$ with $m = 2$, $n_1 = n_2 = n/2$, $p_{1,1} = p_{2,2} = p$, $p_{1,2} = q$.

It can be proved that if $(p-q)^{-1} = o(\sqrt[4]{n/\log n})$ (e.g. if p, q are constant), the partition (V_1, V_2) is almost surely the unique smallest cut of the graph $\mathcal{R}_{n,p,q}$ into two equal parts.

Almost all proofs in the paper are based on the well known Chernoff bound to the tail of the binomial distribution. We will use it in the following form:

Theorem 2.3 *Let $\mathbf{f}_1, \ldots, \mathbf{f}_r$ be independent 0,1-valued random variables. Denote $p_i =$ $\mathbf{Prob}(\mathbf{f}_i = 1)$ for $i = 1, \ldots, r$, $\mathbf{F} = \mathbf{f}_1 + \cdots + \mathbf{f}_r$, $P = p_1 + \cdots + p_r$. Then*

$$\mathbf{Prob}(\mathbf{F} - P \geq \varepsilon P) \leq \exp(-\frac{\varepsilon^2}{3}P), \quad \mathbf{Prob}(P - \mathbf{F} \geq \varepsilon P) \leq \exp(-\frac{\varepsilon^2}{3}P).$$

Let us now give some simple results on our distributions.

Lemma 2.4 *With probability $1-\exp(-\Omega(n/k))$, the classes $V_1, \ldots, V_k$ of the construction $\mathcal{G}_{n,p,k}$ are such that*

$$\frac{3n}{4k} \leq |V_i| \leq \frac{5n}{4k} \qquad \text{for } i = 1, \ldots, k.$$

Proof: Let i be fixed. The probability that a vertex $x \in V_i$ will belong to V_i is $1/k$, and therefore the Chernoff bound implies that

$$\mathbf{Prob}\left(||V_i| - \frac{n}{k}| \geq \frac{1}{4}\frac{n}{k}\right) = \exp(-\frac{1}{50}\frac{n}{k})$$

for large n.

The probability to be estimated is at most k times greater, and therefore equal to $\exp(-\Omega(n/k))$ as well. ♣

Lemma 2.5 *For both distributions $\mathcal{C}$, $\mathcal{R}$ and for any choice of parameters there exists $\vartheta > 0$ such that the probability that $\Delta(x,y) > \vartheta n$ for each x, y is $1 - \exp(-\Omega(n/m))$..*

Proof: is obvious for the cut model, and it follows from Lemma 2.4 for the distribution $\mathcal{C}_{n,p,k}$. ♣

Lemma 2.6 $|E(\deg(x,y)) - \Delta(x,y)| = O(1)$.

Proof: The only reason why the equality $E(\deg(x,y)) = \Delta(x,y)$ is not valid is that the probability that x is connected with itself is not $p_{i,i}$, but 0, and threfore the difference of the numbers is $p_{i,i}p_{i,j} + p_{j,j}p_{j,i}$, where we suppose that $x \in V_i$, $y \in V_j$. ♣

Lemma 2.7 *Let A be a subset of V, and $\mathbf{X}$ be a random s-element subset of V, $s = o(|V|)$. The probability that the intersection of A and X has at least $|A|s/(2|V|)$ elements is at least $1 - \exp(-|A|s/(20|V|))$ for large $|V|$.*

Proof: The choice of $\mathbf{X}$ can done by choosing vertices $\mathbf{x}_1, \ldots, \mathbf{x}_s$ such that $\mathbf{x}_i$ is chosen uniformly from $V - \{\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}\}$. The probability that $\mathbf{x}_i \in A$, conditioned on $|A \cap \mathbf{X}| \leq |A|s/(2|V|)$, is at least $(|A| - |A|s/(2|V|))/|V| \geq (9/10)|A|/|V|$ for large $|V|$ independently of the choice of $\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}$, and therefore it follows from the Chernoff bound that

$$\mathbf{Prob}(|A \cap \mathbf{X}| \leq \frac{|A|s}{2|V|}) \leq \mathbf{Prob}(E(|A \cap \mathbf{X}|) - |A \cap \mathbf{X}| \geq \frac{4}{10} E(|A \cap \mathbf{X}|)) \leq$$

$$\leq \exp\left(-\frac{1}{3}\left(\frac{4}{10}\right)^2 \frac{9}{10}\frac{|A|s}{|V|}\right) = \exp\left(-\frac{1}{20}\frac{|A|s}{|V|}\right).$$

♣

3 Idea

Suppose that the partition used in the construction of the graph $\mathcal{G}$ (or $\mathcal{H}$, resp.) has classes $V_1, \ldots, V_m$.

The main idea of algorithm for both our problems is the same:

- Construct first a set Y of vertices such that $|Y \cap V_i| \geq C \log n$ almost surely for all $i = 1, \ldots, m$ and for a sufficiently large C.
- Approximate numbers $\Delta(x, y)$ for all pairs $x, y \in Y$.
- Find a partition $Y_1, \ldots, Y_m$ of Y so that it is almost surely the partition of Y into sets $Y \cap V_i$.
- Given a vertex x that belong to an unknown V_i, approximate $p_{i,j}$, $j = 1, \ldots, m$, by computing the relative frequency of neighbours of x in $Y \cap V_j$, and use this information to determine i such that $x \in V_i$.

Our algorithm will be a probabilistic one, later we will show how to derandomize it to obtain a deterministic method.

Since the partition V_i is originally unknown, the easiest way to construct Y is to chose a sufficiently large random set of vertices. If s is the size of the smallest class among all V_i, and $|Y| \geq 2C(n/s)\log n$, it follows from the Lemma 2.7 that it is very likely that the actual size of all $Y \cap V_i$ is at least $C \log n$.

The next step of the computation is to approximate numbers $\Delta(x, y)$ for $x, y \in Y$. $\Delta(x, y)/n = p + O(1/n)$, where p is the probability that a random element of X is a neighbour of both x and y. p can easily be approximated by independently drawing a sufficiently long sequence $z_1, \ldots, z_R$ of random vertices of X, and dividing the size of the set $\{i \mid z_i \text{ is a neighbour of both } x \text{ and } y\}$ by R.

If $\Delta(x, y)$ are well approximated, it is easy to partition a set Y into $Y \cap V_i$. In both cases (the distributions $\mathcal{C}_{n,p,k}$ and $\mathcal{R}_{n,p,q}$), if $x \in Y \cap V_i$, $y \in Y \cap V_j$, then $\Delta(x, y)$ depends only on whether $i = j$ or $i \neq j$, and it is greater in the former case. Therefore if $\Delta(x, y)$ are known with sufficient (additive) precision, it is possible to determine pairs of vertices of Y, that belong to the same class of the partition $Y \cap V_i$. The parameter R has to be tuned so that the computation is fast (i.e. R small), but the precision of approximation of Δ's is sufficient.

Suppose now that the sets $Y \cap V_1, \ldots, Y \cap V_m$ are known. If Y has been chosen at random, $Y \cap V_j$ is a random subset of V_j, and hence if x belongs to an (unknown) V_i, $\deg(x, Y \cap V_j)$ (that can be easily evaluated) is likely to be close to $p_{i,j}|Y \cap V_j|$. Since $|Y \cap V_j|$ is known, we can approximate $p_{i,j}$ for all j, which makes it possible to determine i such that $x \in V_i$, because for both distributions $\mathcal{C}$ and $\mathcal{R}$ the value of $p_{i,i}$ is substantially different from $p_{i,j}$, $j \neq i$.

4 Algorithms

We first give an algorithm that gives almost surely a good approximation of $\Delta(x,y)$. It uses positive parameters C, R, $\log n \le R \le n/12$, and numbers m, b, S defined as follows

$$m = k, \quad b = \max\left(27, \frac{8}{3\ln\frac{1}{1-p}}\right) \qquad \text{in the case of the distribution } \mathcal{C}_{n,p,k},$$

$$m = 2, \quad b = \max(27, 24(p-q)^{-2}) \qquad \text{in the case of the distribution } \mathcal{R}_{n,p,q},$$

$S = (C+2)bm\ln n$.

Degree estimation
Input: A graph G with the vertex set V and the edge set E.
Output: A set $Y \subset V$ and numbers $\delta(x,y)$ for all x, y.
Remark: Numbers $\delta(x,y)$ are aimed to approximate $\Delta(x,y)$.
begin
Y := a random subset of V with S elements;
for each $x, y \in Y$ **do**
    $d(x,y) := 0$;
for $i := 1$ **to** R **do begin**
    choose a random vertex $z \in V$;
    for each $x, y \in Y$ **do**
        if both $\{x,z\} \in E$ and $\{y,z\} \in E$ **then**
            $d(x,y) := d(x,y) + 1$;
    end;
for each $x, y \in Y$ **do**
    $\delta(x,y) := nd(x,y)/R$;
end.

Throughout this paragraph we suppose that the vertex z is chosen uniformly among all vertices of V. In the next section we will show that the properties of the algorithm do not change too much, if the distribution is only close to the uniform one.

Lemma 4.1 *Running time of* **Degree estimation** *algorithm is* $O(S^2R)$.

Proof: is obvious. ♣

Lemma 4.2 *With probability* $1 - n^{-C}$, $|Y \cap V_i| \ge (3b/8)(C+2)\ln n$ *for each* i *and both distributions* $\mathcal{C}_{n,p,k}$ *and* $\mathcal{R}_{n,p,q}$.

Proof: Let i be fixed. In view of Lemma 2.4, $|V_i| \ge 3n/4m$ with large probability. Lemma 2.7 implies that

$$|Y \cap V_i| \ge \frac{|Y||V_i|}{2n} \ge \frac{(C+2)bm\ln n}{2n}\frac{3n}{4m} \ge (3b/8)(C+2)\ln n > 10(C+1)\ln n$$

with probability $1 - \exp(-|Y||V_i|/20n) \ge 1 - \exp(-(C+1)\ln n) \ge 1 - n^{-(C+1)}$. ♣

It the remaining part of the paper, ϑ will denote the constant from Lemma 2.5.

We are now going to prove that the numbers $\delta(x,y)$ are likely to be good estimations of numbers $\Delta(x,y)$.

Lemma 4.3 *Given $\varepsilon = \varepsilon(n) > 0$, the probability that $|\delta(x,y) - \Delta(x,y)| < \varepsilon n$ for all $x, y \in Y$ is at least $1 - 4n^2 \exp(-\varepsilon^2 \vartheta R/54)$.*

Proof: The probability that the inequality does not hold for some x, y is at most n^2 times greater that the probability that it is not true for a fixed pair of vertices. Let x, y be two fixed vertices.

Since $\deg(x,y)$, obtained as a result of a random construction of the graph, is a random variable with expectation $\Delta(x,y) + O(1)$, which is a sum of independent 0,1-valued random variables, the Chernoff bound implies that

$$\textbf{Prob}\left(|\deg(x,y) - \Delta(x,y)| \geq \frac{\varepsilon}{3}n\right) \leq \textbf{Prob}\left(|\deg(x,y) - \Delta(x,y)| \geq \frac{\varepsilon}{3}\Delta(x,y)\right) \leq$$

$$\leq 2\exp\left(-\frac{\varepsilon^2}{27}\Delta(x,y)\right) \leq 2\exp\left(-\frac{\varepsilon^2\vartheta}{27}n\right),$$

which also implies that, with probability $1 - \exp(-\Omega(n))$, $\deg(x,y) \geq \vartheta n/2$ for all x, y.

Once the graph is constructed and the degree estimation algorithm applied, the probability that $d(x,y)$ increases during one execution of the **for** loop of the algorithm is exactly $\deg(x,y)/n$, and these probabilities are independent for different executions of the loop.

After the execution of the **for** statement

$$\textbf{Prob}\left(|d(x,y) - R\frac{\deg(x,y)}{n}| \geq \frac{\varepsilon}{3}R\right) \leq$$

$$\leq \textbf{Prob}\left(|d(x,y) - R\frac{\deg(x,y)}{n}| \geq \frac{\varepsilon}{3}R\frac{\deg(x,y)}{n}\right) \leq$$

$$\leq 2\exp\left(-\frac{\varepsilon^2}{27}\frac{R}{n}\deg(x,y)\right) \leq 2\exp\left(-\frac{\varepsilon^2\vartheta}{54}R\right),$$

by the Chernoff bound, and therefore

$$\textbf{Prob}(|\delta(x,y) - \Delta(x,y)| < \varepsilon n) \geq \textbf{Prob}(|\delta(x,y) - \Delta(x,y)| < \frac{2}{3}\varepsilon n) \geq 1 - 4\exp(-\frac{\varepsilon^2\vartheta}{54}R).$$

The bound with 2/3 is not used here, but will be used later. ♣

The algorithm enables us to approximate values of $\Delta(x,y)$ for all pairs of vertices of Y. Once Δ-values are approximated, we can use them to determine the partition induced on Y by the partition used during the construction of the graph as follows:

In the model $\mathcal{C}_{n,p,k}$, Δ-value of any pair of vertices that belong to the same class V_i is always greater than Δ-value of any pair connecting different classes. Moreover, there is a gap between these two sets of numbers. If the gap is sufficiently larger than the inaccuracy committed by the **Degree estimation** algorithm, we can distinguish between "monochromatic" and "bichromatic" pairs with large probability. The knowledge of monochromatic pairs gives immediately the partition of Y into sets $Y \cap Y_i$.

Similarly, pairs of vertices of $\mathcal{R}_{n,p,q}$, belonging to the same class of the original cut have greater Δ-values.

The gap between two classes of values of Δ is a function of parameters of a graph construction. Its size implies the necessary precision of the algorithm, described by ε of Lemma 4.3, which in turn gives a lower bound to the value of R. If parameters are such that the bound to R is an order of magnitude greater than n, the algorithm can not be used. (In this case it seems that no known polynomial time algorithm is able to find a

good solution of the problem with sufficiently large probability). If parameters change so that the bound to R decreases, the time complexity of the algorithm decreases as well. In the most favourable cases, logarithmic R is sufficient, which gives rise to a surprisingly low expected time complexity.

Lemma 4.4 *If x, y are two vertices of $C_{n,p,k}$ then, with probability $1 - \exp(-\Omega(n/k))$,*

$$\Delta(x,y) \geq \left(n - \frac{5}{4}\frac{n}{k}\right)p^2 = np^2 - \frac{5}{4}\frac{np^2}{k}$$

if x, y belong to the same class of the partition $(V_1, \ldots, V_k)$, otherwise

$$\Delta(x,y) \leq \left(n - 2\frac{3}{4}\frac{n}{k}\right)p^2 = np^2 - \frac{6}{4}\frac{np^2}{k}.$$

Proof: If x, y belong to the same class, only elements of that class can not be neighbours of x, y. If x, y belong to different classes, two classes are excluded. The rest of the lemma follows from Lemma 2.4. ♣

Definition 4.5 *Given a number t, define by $\overline{E}_t$ the relation on Y such that $x\overline{E}_t y$ if and only if $\delta(x,y) > t$, and by E_t the finest equivalence containing $\overline{E}_t$.*

Lemma 4.6 *Let G be a graph constructed by distribution $C_{n,p,k}$, Y be a set of vertices of G. If $|\delta(x,y) - \Delta(x,y)| < np^2/(12k)$ for all $x, y \in Y$, and $|Y \cap V_i| = \Omega(\log n)$ for $i = 1, \ldots, k$, then, with probability $1 - n^{-\Omega(\log n)}$, there are t_1, t_2 such that*

- *$\overline{E}_{t_1} = \overline{E}_{t_2}$, $t_2 - t_1 \geq np^2/(12k)$,*
- *$x\overline{E}_{t_1} y$ if and only if x and y belong to the same class V_i, and*
- *given a number t, $t < t_1$ if and only if there exists an edge $\{x, y\}$ of G such that $x, y \in Y$ and $xE_t y$.*

Proof: Let E be an equivalence on Y determined by the partition $Y \cap V_1, \ldots, Y \cap V_k$. Put $t_0 = np^2 - (17/12)np^2/k$, $t_2 = np^2 - (16/12)np^2/k$. Define T as the set of all t such that $t = \delta(x,y)$ for some vertices x, y, t_1 be the smallest element of T such that $\overline{E}_{t_1} = \overline{E}_{t_0}$. If t_3 is the largest element of T such that $t_3 \leq t_0$, then $\overline{E}_{t_3} = \overline{E}_{t_0}$, which implies that $t_1 \leq t_3 \leq t_0$, and therefore $t_2 - t_1$ is sufficiently large. It follows from Lemma 4.4 that, with probability $1 - \exp(-\Omega(n/k)) \geq 1 - n^{-\Omega(\log n)}$, $\overline{E}_{t_1} = \overline{E}_{t_0} = \overline{E}_{t_2} = E$. If t is such that $x\overline{E}_t y$ for some edge $\{x, y\}$, then $t < \delta(x,y) \leq t_1$, because x, y are not equivalent in $E = E_{t_1}$. Now, it is sufficient to prove the rest of the lemma for t, which is the largest element of T smaller than t_1. E_t is coarser than $E = E_{t_1}$, and therefore there exists an equivalence class of E_t containing two different classes V_i, V_j. There are $\Omega(\log^2 n)$ pairs (x, y), $x \in Y \cap V_i$, $y \in Y \cap V_j$, and the probability that no one of them is an edge is $p^{|Y \cap V_i||Y \cap V_j|} = \exp(-\Omega(\log^2 n)) = n^{-\Omega(\log n)}$. ♣

The last lemma shows how to recover the partition of the set Y induced by the original partition of $C_{n,p,k}$: if $\Delta(x,y)$ are approximated, compute $E_{t_1} = E_{t_2}$. The only problem is how to find the treshold t between t_1 and t_2. However, Lemma 4.6 suggests the solution. If two endpoints of an edge of the graph are equivalent in E_t, t is too small, and it should be increased, otherwise we can try to decrease it. Using interval halving, this method converges to an element of $[t_1, t_2]$ after at most $\log_2 n$ iterations.

In order to formulate the next algorithm in the way that can be used for the cut problem as well, we introduce the next definition:

Definition 4.7 *When dealing with the distribution $C_{n,p,k}$, we say that an equivalence E on a set Y of vertices is coarse if there exists an edge $\{x,y\}$ of the graph such that xEy.*

When speaking about $\mathcal{R}_{n,p,q}$, we say that E is coarse, if it has just one equivalence class.

Sample set partition
Input: A graph G.
Output: Sets Y_i of vertices of G, $i = 1, \ldots, k$.
begin
construct a set Y and compute $\delta(x,y)$, $x,y \in Y$ by **Degree estimation**;
$L := -1$; $U := n$;
while $L < U - 1$ **do begin**
    $T := \lfloor (L+U)/2 \rfloor$;
    if E_T is coarse **then** $L := T$ **else** $U := T$;
    end;
return equivalence classes of E_U
end

Theorem 4.8 *Let C be a constant. There is a constant $c > 0$ and a number $R = R(n)$ such that if $k \leq c\sqrt{n/\ln n}$, the time complexity of the algorithm* **Sample set partition** *applied to $C_{n,p,k}$, and using R as one of the parameters, is $O(k^4 \log^3 n)$, and the probability of an incorrect answer is $O(n^{-C})$.*

Proof: Put $\varepsilon = p^2/(12k)$, $R = \lfloor 7777k^2(C+2)p^{-4}\vartheta^{-1}\ln n \rfloor$.

It is easy to check that $R \geq 54(C+2)\varepsilon^{-2}\vartheta^{-1}\ln n$, and therefore $\varepsilon^2\vartheta R/54 \geq (C+2)\ln n$. Lemma 4.3 implies that $|\delta(x,y) - \Delta(x,y)| < \varepsilon n = np^2/(12k)$ for all x,y with probability $1 - O(n^{-C})$. Let t_1, t_2 be numbers from Lemma 4.6. During the computation of **Sample set partition** the equivalence E_L is always coarse (provided the graph has at least one edge, which is true with probability $1 - \exp(-\Omega(n^2))$), E_U is never coarse, and therefore $L < t_1 \leq U$. Since $U - L \leq 1$ at the end of the computation, it follows that final value of U almost surely verifies $t_1 \leq U < t_2$, and therefore E_U is likely to be the correct answer.

Since $R = O(k^2 \log n)$, $S = O(k \log n)$ the time bound to the computation of **Degree estimation** follows from Lemma 4.1. The **while** loop is executed at most $\log_2 n$ times. The relation $\overline{E}_T$ can be found in $O(|Y|^2)$ time, equivalence classes of E_T are connected components of the graph $(Y, \overline{E}_T)$, and therefore E_T can also be found in time $O(|Y|^2)$. It follows that the **while** statement is finished in $O(|Y|^2 \log n) = O(k^2 \log^3 n)$ steps.

Finally R should be less than $n/12$, which implies that we have to suppose $k \leq c\sqrt{n/\ln n}$, where $c = p^2(12 \cdot 7777(C+2))^{-1/2}$.

Note that the large constants in this proof could be considerably improved by being more careful in proofs of the preceeding lemmas. ♣

Now let us turn to the cut problem. It is obvious that

Lemma 4.9 *If x,y are two vertices of $\mathcal{R}_{n,p,q}$ then $\Delta(x,y) = \frac{n}{2}(p^2+q^2)$ if x,y belong to the same class of the partition $(V_1, \ldots, V_k)$, otherwise $\Delta(x,y) = npq$.*

The difference of the two bounds of the lemma is $n(p-q)^2/4$, and therefore we need to approximate $\Delta(x,y)$ with (additive) precision $O(n(p-q)^2)$. Again, the original partition of vertices of $\mathcal{R}_{n,p,q}$ is given by the equivalence classes of $E_{t_1} = E_{t_2}$, where $[t_1, t_2]$ contains in the middle third of the interval between npq and $\frac{n}{2}(p^2+q^2)$. The

convergence process that finds an element of $[t_1, t_2]$ is based on the observation that if t is too small, the equivalence E_t is strictly coarser than the original cut, i.e. it has just one equivalence class, which means that it is coarse in the sense of Definition 4.7.

Theorem 4.10 *Let C be a constant. There are constants $D > 0$ and $R = R(n)$ such that if $p - q \geq D\sqrt{n^{-1}\log n}$, the time complexity of the algorithm* **Sample set partition**, *applied to $\mathcal{R}_{n,p,q}$, and using R as a parameter, is $O((p-q)^{-6}\log^3 n)$, and the probability of an incorrect answer is $O(n^{-C})$.*

Proof: We have to choose R so that $|\delta(x,y) - \Delta(x,y)|$ are smaller than $n(p-q)^2/6$, which means that the constant ε of Lemma 4.3 must be at most $(p-q)^2/6$. In view of the same lemma, $\varepsilon^2\vartheta R/54 \geq (C+2)\ln n$ in order to guarantee the error bound, i.e. $R \geq 54(C+2)\varepsilon^{-2}\vartheta^{-1}\ln n$. It is sufficient to choose $R = \lfloor 1945(C+2)(p-q)^{-4}\vartheta^{-1}\ln n \rfloor$.

The equivalences E_T on Y can be found in time $O(|Y|^2) = O(S^2)$, and therefore the rest follows from Lemma 4.1.

The bound $R \leq n/12$ implies that we have to suppose $p - q \geq D\sqrt[4]{n^{-1}\log n}$, where D is a number that depend only on C and p. ♣

From now on, denote the classes determined by the preprocessing algorithm by $W_1, \ldots, W_m$. The queries of the type "Do vertices x, y belong to the same class V_i of $\mathcal{G}_{n,p,k}$?" are answered by the next algorithm

Query
Input: A graph G with the vertex set V and the edge set E,
sets $W_1, \ldots, W_m$,
and vertices $x, y \in V$.
begin
$\varphi(x) = i$, where i minimize the number $\deg_{W_i}(x)/|W_i|$;
$\varphi(y) = j$, where j minimize the number $\deg_{W_j}(y)/|W_j|$;
if $\varphi(x) = \varphi(y)$
then answer "x and y are in the same class"
else answer "x and y are not in the same class";
end.

The same algorithm can be applied to the distribution $\mathcal{R}_{n,p,q}$, however we need to define the value of the function φ as i (j, resp.) that *maximize* $\deg_{W_i}(x)/|W_i|$ ($\deg_{W_j}(y)/|W_j|$, resp.).

Now we are going to prove that the any vertex x belongs almost surely to the class $V_{\varphi(x)}$ of the original partition.

Lemma 4.11 *Let G be a graph constructed by the distribution $\mathcal{G}_{n,p,k}$, Y be the set constructed by* **Degree estimation** *algorithm. Denote $\kappa(x,j) = \deg_{Y\cap V_j}(x)/|Y \cap V_j|$ for each vertex x and $j = 1, \ldots, m$. The probability that $\kappa(x,i) < \kappa(x,j)$ for each x, i, j such that $x \in V_i$, $x \notin V_j$ is $1 - O(n^{-C})$.*

Proof: The size of all $Y \cap V_i$ is at least $(3b/8)(C+2)\ln n$ with probability $1 - O(n^{-C})$, see Lemma 4.2.

Fix $j \neq i$ and $x \in V_i$. $\kappa(x,i) = 0$, and the probability that there is no edge between x and $Y \cap V_j$ is at most

$$(1-p)^{|Y\cap V_j|} \leq (1-p)^{(3b/8)(C+2)\ln n} = \exp(-(3b/8)(C+2)\ln\frac{1}{1-p}\ln n) = n^{-(C+2)}.$$

The probability that there are x, j, such that $x \notin V_j$ and $\kappa(x,j) = 0$ is at most n^2 times greater. ♣

Lemma 4.12 *Let G be a graph constructed by the distribution $\mathcal{R}_{n,p,q}$, Y be the set constructed by* **Degree estimation** *algorithm. Denote $\kappa(x,j) = \deg_{Y\cap V_j}(x)/|Y \cap V_j|$ for each vertex x and $j = 1,\ldots,m$. The probability that $\kappa(x,i) > \kappa(x,j)$ for each x,i,j such that $x \in V_i$, $x \notin V_j$ is $1 - O(n^{-C})$.*

Proof: The sizes of both $|Y \cap V_1|$ and $|Y \cap V_2|$ are at least $(3b/8)(C+2)\ln n$ with probability $1 - O(n^{-C})$, see Lemma 4.2.

Now, let x,i,j be fixed, $x \in V_i$, $x \notin V_j$.

$$\mathbf{Prob}(\frac{p+q}{2} \geq \kappa(x,i)) = \mathbf{Prob}(p - \kappa(x,i) \geq \frac{p-q}{2}) =$$

$$= \mathbf{Prob}(p|Y \cap V_i| - \kappa(x,i)|Y \cap V_i| \geq \frac{p-q}{2}|Y \cap V_i|) \leq$$

$$\leq \exp(-\frac{(p-q)^2}{12}\frac{S}{4}) \leq \exp\left(-\frac{(p-q)^2(C+2)2b\ln n}{48}\right) \leq$$

$$\leq \exp(-(C+2)\ln n) = n^{-(C+2)},$$

$$\mathbf{Prob}(\kappa(x,i) \geq \frac{p+q}{2}) = \mathbf{Prob}(\kappa(x,j) - q \geq \frac{p-q}{2}) =$$

$$= \mathbf{Prob}(\kappa(x,j)|W_j| - q|Y \cap V_j| \geq \frac{p-q}{2}|Y \cap V_j|) \leq \exp(-\frac{(p-q)^2}{12}\frac{S}{4}) \leq n^{-(C+2)},$$

and therefore $\mathbf{Prob}(\kappa(x,i) \leq \kappa(x,j)) \leq 2n^{-(C+2)} = O(n^{-(C+2)})$.

The probability that there are x, j such that $x \notin V_j$, and $\kappa(x,i) \leq \kappa(x,j)$, where $x \in V_i$, is at most n^2 times greater. ♣

As a consequence, we obtain

Lemma 4.13 *If C is a constant, we can choose the parameter R so that the running time of algorithms* **Degree estimation** *and* **Sample set partition** *is $O(k^4\log^3 n)$ for the coloring problem and $O((p-q)^{-6}\log^3 n)$ for the cut problem, the running time of* **Query** *is $O(k\log n)$ for the coloring problem, and $O((p-q)^{-2}\log n)$ for the cut problem, and, with probability $1 - O(n^{-C})$, a graph constructed by the distribution $\mathcal{G}_{n,p,k}$, $\mathcal{R}_{n,p,q}$, resp. is such that the computation of* **Degree estimation**, **Sample set partition**, *and* **Query** *for any pair of vertices is correct.*

5 Derandomization

The algorithm **Degree estimation** is randomized. This section shows that the same results with respect to the partitions $\mathcal{C}_{n,p,k}$ and $\mathcal{R}_{n,p,q}$ can be obtained by a deterministic method.

Throughout this section, we will suppose that the set of vertices of the graph is $V = \{1,\ldots,n\}$.

First, we show that the random set Y of the algorithm **Degree estimation** can be replaced by any fixed set of S vertices, e.g. by the set $Y = \{1,\ldots,S\}$.

The choice of a random Y of the algorithm **Degree estimation** can be implemented as follows: choose a random permutation π of V and put $Y := \{\pi(1),\ldots,\pi(S)\}$. The probability that the computation is correct is the same if the algorithm is applied to $\mathcal{C}_{n,p,k}$ ($\mathcal{R}_{n,p,q}$, resp.) with randomly chosen set Y, or if we use $\pi^{-1}(Y) = \{1,\ldots,S\}$ and the input graph $\pi^{-1}(\mathcal{C}_{n,p,k})$ ($\pi^{-1}(\mathcal{R}_{n,p,q})$,resp.), i.e. the graph with edges $\{\pi^{-1}(x),\pi^{-1}(y)\}$ for any edge $\{x,y\}$ of the original graph. However, the distribution $\pi^{-1}(\mathcal{C}_{n,p,k})$ is equal to $\mathcal{C}_{n,p,k}$, similarly for $\mathcal{R}_{n,p,q}$.

From now on, let us suppose that the set Y of **Degree estimation** is $\{1,\ldots,S\}$. The crucial observation is that if $x,y \in V-Y$, our algorithms never ask whether x and y are connected by an edge. Let $b(i,j)$ is the bit that indicates whether vertices i and j are connected by an edge ($b(i,j)=1$) or not ($b(i,j)=0$). Values of bits $b(i,j), i,j > S$ are random variables that do not influence the result of the computation and therefore they can be used as a source of randomness for the randomized degree estimation. If the input graph is generated by either $\mathcal{C}_{n,p,k}$ or $\mathcal{R}_{n,p,q}$, all these bits are independent. However, the probability that $b(i,j)=1$ is not $1/2$ in general, in some cases (when i and j are in the same class of the partition in the case of coloring) it might even be always equal to 0.

In order to cope with imperfect random bits, we will group them into larger classes, and use sums modulo 2 of groups instead of bits itself. Such bits are independent, too, and the probabilities that they are equal to 1 are so close to $1/2$, that the results of previous sections hold without change (though we have to modify proofs). In this way we can obtain $O(n^2/\log n)$ good random bits. Since we need $R\log_2 n = O(n\log n)$ bits only, we will choose size of groups of bits large to improve the quality of bits.

Put $r=\lfloor n/4\rfloor$, $s=\lceil\log_2 n\rceil$, $t=\lfloor n/(4s)\rfloor$. Recall that we suppose $R \le n/12$.

Lemma 5.1 *Let $0<p<1$ be a constant, $b_1,\ldots,b_q$ be independent random boolean variables such that* $\mathbf{Prob}(b_i=1)=p$ *for all i. Then*

$$\mathbf{Prob}(b_1 \oplus \ldots \oplus b_q = 1) = \frac{1}{2}(1+\exp(-\Omega(q))).$$

Proof: Without loss of generality we can suppose that $p \le 1/2$. Put $\pi = 1-2p$. We will show by induction that $\mathbf{Prob}(b_1 \oplus \ldots \oplus b_i = 1) = \frac{1}{2}(1-\pi^i)$.

The equality holds for $i=1$,

$$\mathbf{Prob}(b_1 \oplus \ldots \oplus b_{i+1} = 1) =$$

$$= \mathbf{Prob}((b_1 \oplus \ldots \oplus b_i = 0) \wedge (b_{i+1}=1)) + \mathbf{Prob}((b_1 \oplus \ldots \oplus b_i = 1) \wedge (b_{i+1}=0)) =$$

$$= \frac{1}{2}(1+\pi^i)\frac{1}{2}(1-\pi) + \frac{1}{2}(1-\pi^i)\frac{1}{2}(1+\pi) =$$

$$= \frac{1}{4}(1-\pi+\pi^i-\pi^{i+1}) + \frac{1}{4}(1+\pi-\pi^i-\pi^{i+1}) = \frac{1}{2}(1-\pi^{i+1}).$$

♣

Definition 5.2 *For $a=1,\ldots,r$, $b=1,\ldots,s$ put*

$$B_{a,b} = b(2r+a, 3r+(b-1)t+1) \oplus \ldots b(2r+a, 3r+bt).$$

Probabilities $\mathbf{Prob}(B_{a,b}=1)$ depend only on the partition $V_1,\ldots,V_m$ of vertices of the graph $\mathcal{G}$ or $\mathcal{H}$, and we show that

Lemma 5.3 *With probability $1-\exp(-\Omega(n/\log n)))$, a random partition of the vertex set of either $\mathcal{C}_{n,p,k}$ or $\mathcal{R}_{n,p,q}$ is such that*

$$\mathbf{Prob}(B_{a,b}=1) = \frac{1}{2}(1+\exp(-\Omega(n/\log n))))$$

for all a,b.

Proof: Fix a, b, put $x = 2r + a$, $y_i = 3r + (b-1)t + i$. Since the class of the partition containing x is a random set of size at most $2n/3$ (see Lemma 2.4), the size of the set A of all i such that x, y_i belong to different classes of the partition is at least $n/4$ with probability $1 - \exp(-\Omega(n)))$, see the Chernoff bound. Let us fix values of $b(x, y_i)$ for $i \notin A$, and denote by w their sum modulo 2. Since $B_{a,b} = 1$ iff the sum modulo 2 of all $b(x, y_i,\ i \in A$ is unequal to w, Lemma 5.1 implies that $\mathbf{Prob}(B_{a,b} = 1) = (1/2)(1 + \exp(-\Omega(t)))$. ♣

Let $\mathbf{x}_a$, $a = 1, \ldots, r$ be the number with binary digits $B_{a,1} \ldots B_{a,s}$. Since $B_{a,1} = 0$ implies that $\mathbf{x}_a \leq 2^{s-1} \leq n$, the probability that $\mathbf{x}_a \in V$ is at least $(1/2)(1 + \exp(-\Omega(n/\log n)))$, and therefore it follows from the Chernoff bound that, with probability $1 - \exp(-\Omega(r)) = 1 - \exp(-\Omega(n))$, at least $r/3 \geq R$ vertices $\mathbf{x}_a$ belong to V, which means that we have almost surely enough random vertices to feed the algorithm **Degree estimation.**

Now we prove that the distributions $\mathbf{x}_a$ are almost uniform.

Lemma 5.4

$$\mathbf{Prob}(\mathbf{x}_a = x \mid \mathbf{x}_a \in V) = \frac{1}{n}(1 + \exp(-\Omega(n/\log n)))$$

for $a = 1, \ldots, r$ and each $x \in V$.

Proof: It follows from Lemma 5.3 that

$$\mathbf{Prob}(\mathbf{x}_a = x) = \left[\frac{1}{2}(1 + \exp(-\Omega(n/\log n)))\right]^s =$$

$$= 2^{-s}(1 + s\exp(-\Omega(n/\log n))) = 2^{-s}(1 + \exp(-\Omega(n/\log n))),$$

$$\mathbf{Prob}(\mathbf{x}_a \leq n) = n2^{-s}(1 + \exp(-\Omega(n/\log n))),$$

and therefore the conditional probability to be estimated is

$$\mathbf{Prob}(\mathbf{x}_a = x \mid \mathbf{x}_a \leq n) = \frac{2^{-s}(1 + \exp(-\Omega(n/\log n)))}{n2^{-s}(1 + \exp(-\Omega(n/\log n)))} = \frac{1}{n}(1 + \exp(-\Omega(n/\log n))).$$

♣

The only part of the proof of the correctness of the algorithms of Paragraph 4 that has to be modified is the estimation of the probability that $\delta(x, y)$ and $\deg(x, y)$ are close in the proof of Lemma 4.3. Now, the probability that a random vertex $\mathbf{x}_a$ hits the set of $\deg(x, y)$ common neighbours of x and y is not exactly $\deg(x, y)/n$, but

$$\sum_z \mathbf{Prob}(\mathbf{x}_a = z \mid \mathbf{x}_a \in V) = \frac{\deg(x, y)}{n}(1 + \exp(-\Omega(n/\log n))),$$

where the sum is over all common neighbours z of x and y. This means that, with probability $1 - 2\exp(-\varepsilon^2 \vartheta R/55)$ determined in Lemma 4.3,

$$|\delta(x, y) - \deg(x, y)(1 + \exp(-\Omega(n/\log n)))| \leq \frac{\varepsilon n}{3} \text{ and } |\deg(x, y) - \Delta(x, y)| \leq \frac{\varepsilon n}{3}.$$

However,

$$|\deg(x, y)(1 + \exp(-\Omega(n/\log n))) - \deg(x, y)| = \exp(-\Omega(n/\log n))\deg(x, y) \leq \frac{\varepsilon n}{3}$$

for large n, which means that the difference of $\delta(x, y)$ and $\Delta(x, y)$ is still likely to be less than εn.

We can therefore reformulate main results of Paragraph 4 as follows:

Theorem 5.5 *Let C be a constant, and $k = c\sqrt{n/\log n}$ for sufficiently small constant c. There exists a deterministic algorithm which, applied to $\mathbf{G} = \mathcal{C}_{n,p,k}$, answers queries "Given vertices x, y, is there a k-coloring of $\mathbf{G}$ such that x, y receive the same color?" in time $O(k \log n)$ with preprocessing time $O(k^4 \log^3 n)$; the probability that some answer is incorrect is bounded by $O(n^{-C})$.*

Theorem 5.6 *Let C be a constant, and $p - q \geq D\sqrt{\log n/n}$ for some sufficiently large constant D. There exists a deterministic algorithm which, applied to $\mathbf{G} = \mathcal{R}_{n,p,q}$, answers queries "Do given vertices x, y belong to the same class of (some) minimum cut of $\mathbf{G}$ into equal parts?" in time $O(\log n)$ with preprocessing time $O((p-q)^{-2} \log^2 n + \log^3 n)$; the probability that some answer is incorrect is bounded by $O(n^{-C})$.*

If the function φ of the algorithm **Query** is evaluated for all vertices, the sets $\varphi^{-1}(i)$, $i = 1, \ldots, m$ give the partition $V_1, \ldots, V_m$ with probability $1 - O(n^{-C})$. It follows the time that is equal to the preprocessing time plus n times the query time is sufficient to obtain a partition of vertices such that, with probability $1 - O(n^{-C})$, it is either k coloring of $\mathcal{C}_{n,p,k}$ or an $(qn^2/4)$-cut of $\mathcal{R}_{n,p,q}$.

In the case of the coloring problem, we can check whether the result is really a k-coloring using an $O(n^2)$ worst-case and $O(n^2/k)$ expected-case time procedure that first sorts vertices by colors and then checks all monochromatic pairs of vertices. If k is not too large, we are in a paradoxial situation when it is faster to obtain a solution that is almost surely correct than to verify its correctness.

In the unlikely case when the result is not a legal k-coloring, we can apply slower but more reliable algorithms of Dyer and Frieze [3], thus obtaining a $O(n^2/k)$ expected time algorithm that produces always a k coloring (the average time necessary for the failure processing is $o(1)$ in view of small probability that the basic algorithms fails). If k is not constant, this time is sublinear with respect to the expected number of edges of the graph $\mathcal{C}_{n,p,k}$.

A similar approach can be used for the cut problem, where we can even guarantee that the optimal equitable cut of $\mathcal{R}_{n,p,q}$ is found. E.g. if $q < p$ are constants, our approach gives an $O(n \log n)$ algorithm that is very likely to produce the optimal cut. Unfortunately, we need quadratic time (both expected and worst-case) to count the number of the cross edges, and cubic time to prove the optimality of the cut [3]. Hence the verification is again much slower than the computation.

6 Other graph problems and distributions

The methods presented in the paper apply to a wider collection of graph theoretic problems. As an illustration, we indicate how they can be used to find a large independent subset or clique of a graph.

The distributions used in this paragraph will be denoted by $\mathcal{Q}_{n,p,k}$ and $\overline{\mathcal{Q}}_{n,p,k}$, and are defined as the distribution $\mathcal{H}$ with parameters $m = 2$, $n_1 = k$, $n_2 = n - k$, $p_{1,1} = 0$, $p_{2,2} = p$, and $p_{1,2} = p$ for $\mathcal{Q}_{n,p,k}$, $p_{1,2} = p(n-k-1)/(n-2k)$ for $\overline{\mathcal{Q}}_{n,p,k}$.

The construction of $\mathcal{Q}_{n,p,k}$ can be reformulated as follows. Construct first a random graph $\mathbf{G}_{n,p}$ with the constant edge probability p, then choose a random set A of k vertices, and remove all edges inside A. If $k \geq C\sqrt{n \log n}$ C large, then the graph $\mathcal{Q}_{n,p,k}$ is almost surely such that any vertex of A has smaller degree than arbitrary vertex of $V - A$, because the expectations of degrees are $(n-k)p$ inside A and $(n-1)p$ outside A, while the standard deviation of degrees are $O(\sqrt{np(1-p)}) = o((k-1)p)$. Thus, for moderately large k it is quite simple to find the set A almost surely. On the other hand, if $k = o(\sqrt{n \log n})$, no polynomial time algorithm is known to find an

independent set of size k with sufficiently large probability. In order to make the task for larger k more difficults, we modify the construction $\mathcal{Q}_{n,p,k}$ to $\overline{\mathcal{Q}}_{n,p,k}$ by increasing the probability $p_{1,2}$ so that the expectations of degrees of all vertices are the same.

It is not difficult to prove that for $k \geq C\sqrt{n \log n}$ the set V_1 is almost surely the unique largest independent subset of both $\mathcal{Q}_{n,p,k}$ and $\overline{\mathcal{Q}}_{n,p,k}$. The queries "Does a given vertex x belong to the largest independent set of the graph?" can be answered almost surely correctly as follows:

Apply the algorithm **Degree estimation** with $S = 2C(n/k)\ln n$ to $\mathcal{Q}_{n,p,k}$ or $\overline{\mathcal{Q}}_{n,p,k}$ to produce the sample set Y and to find approximations of $\Delta(x,y)$ for $x, y \in Y$. The value of S guarantees that $V_1 \cap Y$ is likely to be at least $C \ln n$.

The algorithm that finds the partition of Y into $Y \cap V_1$ and $Y \cap V_2$ is different for the two distributions. In the case of $\mathcal{Q}_{n,p,k}$,

$$\Delta(x,y) = (n-k)p^2 \text{ if either } x \in V_1 \text{ and/or } y \in V_1,$$

$$\Delta(x,y) = np^2 \text{ for } x \in V_2, y \in V_2,$$

while in the case of $\overline{\mathcal{Q}}_{n,p,k}$

$$\Delta(x,y) = np^2\left(1 + \frac{k}{n} + o\left(\frac{k}{n}\right)\right) \text{ for } x \in V_1, Y \in V_1,$$

$$\Delta(x,y) = np^2\left(1 + o\left(\frac{k}{n}\right)\right) \text{ if either } x \in V_2 \text{ or } Y \in V_2.$$

Let W_t be the set of all vertices $x \in Y$ such that $\delta(x,y) < t$ (the distribution $\mathcal{Q}_{n,p,k}$) or $\delta(x,y) > t$ (the distribution $\overline{\mathcal{Q}}_{n,p,k}$) for some $y \in Y$. If $t = np^2 - kp^2/2$, $t = np^2 + kp^2/2$, resp., the set V_1 is almost surely the set W_t. The algorithm similar to **Sample set partition** can be based on the observation that if t is such that W_t is larger than V_1, then it is very likely that W_t contain at least one edge.

Finally, if $Y \cap V_1$ is known, the query algorithm uses the fact that $x \in V_1$ implies $\deg(x, Y \cap V_1) = 0$, while in the case $x \in V_2$ the value of $\deg(x, Y \cap V_1) = 0$ is almost surely quite large, i.e. not equal to zero.

Therefore it is possible to prove that

Theorem 6.1 *Let C be a constant. There is a constant $D > 0$ such that queries "Does a vertex x belong to the largest independent set?" on graphs $\mathcal{Q}_{n,p,k}$ or $\overline{\mathcal{Q}}_{n,p,k}$ can be answered deterministically in time $O((n/k)\log n)$ with preprocessing in time $O((n/k)\log n)$, and with probability of an incorrect answer bounded by $O(n^{-C})$, provided $k \geq D\sqrt{n \log n}$.*

The probability distribution given in paragraph 2 is based on the model with independent edge probabilities. It is well known that the model with edge probability p has very similar properties to the model with random set of $p\binom{n}{2}$ edges. Based on this observation, we can modify distributions $\mathcal{G}$ and $\mathcal{H}$ to distributions $\mathcal{G}^*$, $\mathcal{H}^*$ as follows:

Definition 6.2 *Let n, m, $p_{i,j}$, n_i, V_i be the same as in Definition 2.2. Let $A_{i,j}$ be collection of all pairs $\{x,y\}$ such that $x \in V_i$, $y \in V_j$, $x \neq y$, and $\mathbf{E}_{i,j}$ be a set chosen uniformly at random from the collection of all subsets of $A_{i,j}$ with $\lfloor p_{i,j}|V_i||V_j| \rfloor$ elements in the case $i \neq j$, and $\lfloor p_{i,i}\binom{|V_i|}{2} \rfloor$ elements if $i = j$.*

The graph with the vertex set V and edge set $\bigcup_{i,j} \mathbf{E}_{i,j}$ is denoted by $\mathcal{G}^*$, if the sets V_i were constructed as in the definition of $\mathcal{G}$, or by $\mathcal{H}^*$, if V_i were obtained as in $\mathcal{H}$.

We define $\mathcal{C}^*_{n,p,k}$, $\mathcal{R}^*_{n,p,q}$, $\mathcal{Q}^*_{n,p,k}$, and $\overline{\mathcal{Q}}^*_{n,p,k}$ similarly to $\mathcal{C}_{n,p,k}$, $\mathcal{R}_{n,p,q}$, $\mathcal{Q}_{n,p,k}$, $\overline{\mathcal{Q}}_{n,p,k}$, using $\mathcal{G}^*$, $\mathcal{H}^*$ instead of $\mathcal{G}$, $\mathcal{H}$.

Graphs $\mathcal{G}$ ($\mathcal{H}$, resp.) and $\mathcal{G}^*$ ($\mathcal{H}^*$, resp.) correspond to Model 1 and Model 2 of Dyer and Frieze [3].

Our proofs rely only on the fact that the distribution of $\deg(x,y)$ for two vertices of $\mathcal{G}$ or $\mathcal{H}$ is concentrated around its expectation, which is between $\Delta(x,y)-2$ and $\Delta(x,y)$, and if $x \in V_i$, $A \subset V_j$, $x \notin A$, then $\deg(x,Y)$ is concentrated around its expectation $p_{i,j}|A|$. Both properties hold for $\mathcal{G}^*$, $\mathcal{H}^*$ as well, see [3], and therefore our results remain valid for $\mathcal{C}^*_{n,p,k}$ and $\mathcal{R}_{n,p,q}$.

Note that $\mathcal{R}^*_{n,p,q}$ has always a cut into two equal parts of size exactly $\lfloor qn^2/2 \rfloor$ edges. If $0 < q < p < 1$ are constants, it is proved in [3] that the cut into V_1 and V_2 is the unique optimal cut into equal parts, and that the distribution $\mathcal{R}^*_{n,p,q}$ approximates the uniform distribution on the class of all graphs that have a cut of this size. In view of these results, Theorem 1.2 is a consequence of Theorem 5.6, and similar results are valid also for distributions $\mathcal{C}_{n,p,k}$, $\mathcal{Q}_{n,p,k}$, and $\overline{\mathcal{Q}}_{n,p,k}$.

References

[1] Bollobás, B., The chromatic number of random graphs, *Combinatorica* **8**, 49-56.

[2] Chernoff, H., A measure of asymptotic efficiency for tests based on the sum of observations, *Ann. Math. Statist.* **23** (1952), 493-509.

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

[4] Grimmett, G.R., and McDiarmid, C.J.H., On colouring random graphs, *Math. Proc. Cambridge Phil. Soc.*, **77** (1975), 313-324.

[5] Kučera, L., Expected behavior of graph coloring algorithms, in *Foundations of Computation Theory 77, M. Karpinski, ed., Lecture Notes Comput. Sci.* **56** (Springer, Berlin, 1977), 447-451.

[6] Kučera, L., The greedy coloring is a bad probabilistic algorithm, *J.Algorithms*, **12** (1991), 674-684.

[7] Kučera, L., Graphs with small chromatic numbers are easy to color, *Information Processing Letters* **30** (1989), 233-236.

[8] Matula, D., and Kučera, L., An expose-and-merge algorithm and the chromatic number of a random graph, in *Random Graphs'87, M.Karonski, J.Jaworski, A.Rucinski, eds.*, (J.Wiley and Sons, 1990), 175-187.

[9] Turner, J., Almost all k-colorable graphs are easy to color, *J. Algorithms* **9** (1988), 63-82.