London Taught Course Centre: Graph Theory
Exam Solutions
2021

Question 1

(a) The $(m \times n)$-*grid* is the graph with vertex set $\{(x, y)\colon 1 \le x \le m, 1 \le y \le n\}$ and edge set

$$\Big\{(x_1, y_1)(x_2, y_2)\colon \text{ either } |x_1 - x_2| = 1 \text{ or } |y_1 - y_2| = 1 \text{ (but not both)}\Big\}.$$

Determine the list chromatic number of the $(m \times n)$-grid.

Solution: For $m = n = 1$ this clearly is 1 and for $m = 2$ and $n \le 2$ (or vices versa) this can easily be seen to be 2.

In general, the $(m \times n)$-grid is obviously 2-degenerate, so the greedy colouring algorithm (with the vertex order being a degeneracy order) works with lists of length 3. For lists of length 2, on the other hand we cannot always find a proper colouring if $m \ge 2$ and $n > 2$: In this case the $(m \times n)$-grid contains the (2×3)-grid, which is not 2-list colourable as the following choice of lists shows. We give the vertices $(1, 2)$ and $(2, 2)$ both the list $\{1, 2\}$, so in a valid colouring one of these two vertices gets colour 1 and the other colour 2. We give the vertices $(1, 1)$ and $(2, 3)$ both the list $\{1, 3\}$, and the two remaining vertices $(2, 1)$ and $(1, 3)$ both the list $\{2, 3\}$. Now assume first that $(2, 1)$ is coloured 1 and hence $(2, 2)$ is coloured 2. Then for both vertices $(1, 1)$ and $(2, 1)$ only colour 3 is possible, which is not permissible because these two vertices are adjacent. Similarly, if $(2, 1)$ is coloured 2 and hence $(2, 2)$ is coloured 1, then for both vertices $(3, 1)$ and $(3, 2)$ only colour 3 is possible, which again is not permissible. We conclude that the list chromatic number of the $(m \times n)$-grid in the general case is 3.

(b) A *double torus* is a surface that is homeomorphic to the sphere with 2 handles, that is, a double torus is an orientable surface with genus 2. A *tcdt-graph* (two-cell double-toroidal graph) is a (simple) graph which has a 2-cell embedding on a double torus.

State and prove an upper bound on the chromatic number of tcdt-graphs.

Solution: For planar graphs we derived a bound by determining the degeneracy and using greedy colouring. So lets try to do this here too. We first want a bound on the degeneracy of a tcdt-graph G. We take the Euler-Poicaré formula as given: In our case the Euler characteristic is $2 - 2 \cdot 2 = -2$, and hence we have $v - e + f = -2$, if we have a 2-cell embedding (as assumed) of G with v vertices, e edges, and f faces. It is still true that in an embedding, each edge is either contained in the walks around two faces or twice on the walk around one face, and the walk around each face has length at least 3, and thus $3f \le 2e$. Plugging this in, we obtain $-2 = v - e + f \le v - e + 2e/3$ and hence $e \le 3v + 6$. We conclude that for $v > 6$ we have a vertex of degree at most 7: Otherwise we would have $8v \le 2e$ by the handshaking lemma, and hence $4v \le 3v + 6$, which is only true for $v \le 6$, where trivially all our vertices have degree at most 5. We conclude that G is 7-degenerate, and hence we can use the greedy colouring algorithm to colour tcdt-graphs with 8 colours.

Question 2

Let $F_{n,n}$ be a graph with vertex set $\{x_1, \dots, x_n, y_1, \dots, y_n\}$ whose edge set is randomly constructed as follows. For each $1 \le i < j \le n$ independently, we consider the four potential edges

$$x_i x_j\,, \quad x_i y_j\,, \quad y_i x_j\,, \quad y_i y_j\,,$$

pick one uniformly at random that we *omit*, and then add the three remaining ones as edges to $F_{n,n}$.

(a) Prove that for every $\varepsilon > 0$ with probability tending to 1 as n tends to infinity, the graph $F_{n,n}$ does not contain a complete graph K_s with $s \ge \varepsilon n$ as a subgraph.

Solution: A copy of K_s in this graph can only use at most one vertex of every pair x_i, y_i. Hence, such a K_s uses a vertex from exactly s of the pairs, for which there are $\binom{n}{s}$ choices, and in each pair one vertex, for which there are 2^s choices. For a pair of vertices $v_i v_j$ with $v_m \in \{x_m, y_m\}$ for $m \in \{i, j\}$ the probability that $v_i v_j$ forms an edge is $\frac{3}{4}$ and this is independent from all other pairs $v_{i'} v_{j'}$ as long as $\{i, j\} \ne \{i', j'\}$. We conclude that the expected number of K_s in $F_{n,n}$ is $\binom{n}{s} 2^s (\frac{3}{4})^{\binom{s}{2}} \le (2n)^s (\frac{3}{4})^{s^2/3}$, where the inequality holds for large s. For $s = \varepsilon n$ this expectation is at most

$$(2n)^{\varepsilon n} \left(\frac{3}{4}\right)^{\varepsilon^2 n^2/3} = 2^{\varepsilon n \log(2n) - (\varepsilon^2 \log(4/3)/3) n^2}\,,$$

which goes to 0 with n. We conclude that with high probability there is no $K_{\varepsilon n}$.

(b) Prove that for every $\varepsilon > 0$ with probability tending to 1 as n tends to infinity, the graph $F_{n,n}$ does not contain a K_t-minor with $t \ge (\frac{2}{3} + \varepsilon) n$ as a subgraph.

Solution: We use the following characterisation of minors: A K_t-minor in $F_{n,n}$ is a collection of t vertex disjoint bags $B_1, \dots, B_t \subseteq V(F_{n,n})$ such that $F_{n,n}[B_i]$ is connected for each i, and for each $i \ne j$ there is some edge between B_i and B_j in $F_{n,n}$. Part (a) tells us there is no K_t-minor with all bags of size 1 for $t \ge \frac{1}{2} \varepsilon n$. It is also easy to see that there is no K_t-minor with all bags of size at least 3 for $t \ge (\frac{2}{3} + \varepsilon) n$ as $F_{n,n}$ does not have enough vertices for this. It thus remains to check for minors in which all bags are of size at most 2. Assume that such a minor exists, then it has at least $(\frac{2}{3} + \frac{1}{2}\varepsilon) n$ bags of size exactly 2, because otherwise there would be a $K_{\frac{1}{2}\varepsilon n}$-subgraph. Hence, it remains to show that no $K_{t'}$-minor with all bags exactly size 2 exists for $t' \ge (\frac{2}{3} + \frac{1}{2}\varepsilon) n$.

Indeed, in such a minor each B_i contains vertices from two different pairs x_j, y_j and $x_{j'}, y_{j'}$, since pairs do not contain edges. Also, by possibly omitting two thirds of the bags, we obtain a $K_{t''}$-minor with $t'' \ge \frac{2}{3}(\frac{2}{3} + \frac{1}{2}\varepsilon) n \ge \varepsilon n$ with bags $B_1, \dots, B_{t''}$ such that for each pair x_j, y_j only at most one vertex is contained in $\bigcup_i B_i$.

Now, let $B_1, \dots, B_{t''}$ be one possible choice of such bags. Let us estimate the probability that these give a $K_{t''}$-minor. Consider two of the bags $B_i = \{b_i, b_i'\}$ and $B_j = \{b_j, b_j'\}$ with $i \ne j$. By assumption the vertices b_i, b_i', b_j, b_j' come from 4 different pairs of $F_{n,n}$. So the probabilities that $b_i b_j$ is an edge, that $b_i' b_j$ is an edge, that $b_i b_j'$ is an edge, and that that $b_i' b_j'$ is an edge are pairwise independent, and as in (a), each of these probabilities is $\frac{3}{4}$. We conclude that the probability that there is one of these edges is $1 - (\frac{1}{4})^4$. Consequently, the probability that there is an edge between each pair of bags in $B_1, \dots, B_{t''}$ is $\left(1 - (\frac{1}{4})^4\right)^{\binom{t''}{2}}$. For $t'' = \varepsilon n$, since we have at most $\binom{2n}{2\varepsilon n}$ ways to select $2\varepsilon n$ vertices in $F_{n,n}$, one from each pair (though we are ignoring this in the counting), and there are at most $(2\varepsilon n)!$ ways of pairing these vertices up into bags (again, this is overcounting), we get that there are at most $\binom{2n}{2\varepsilon n}(2\varepsilon n)! \leq (2n)^{2\varepsilon n}$ choices for bags. Hence, the probability that we get a $K_{t''}$-minor of the described form is at most $(2n)^{2\varepsilon n}\big(1-(\frac{1}{4})^4\big)^{\binom{\varepsilon n}{2}}$, which tends to 0 as n tends to infinity, similarly as in (a).

This proof is taken from `https://arxiv.org/pdf/2103.10684.pdf`.

Question 3

For U and V disjoint with n vertices each, let (U,V) be an ε-regular pair with density at least d. Solve the following questions "by hand", that is, without using any existing lemmas that directly provide the answers (you may use results along the way that you find, however). A *binary tree* is a tree in which at most 1 vertex has degree 2 and all other vertices have degree 3 or 1.

(a) What is the longest path you can find in (U,V)?

(b) What is the longest cycle you can find in (U,V)?

(c) What is the biggest binary tree you can find in (U,V)?

(d) What is the biggest complete bipartite graph you can find in (U,V)?

Solution: In each of U and in V we could have εn isolated vertices, so we certainly cannot hope for connected subgraphs on more than $2n(1-\varepsilon)$ vertices. So we can only hope for almost spanning connected subgraphs. For the first three parts we indeed always get almost spanning subgraphs (if ε is sufficiently small compared to d).

To see this in the case of paths, we can simply embed them greedily as follows: Choose a vertex x_1 in U at least $\frac{d}{2}n$ neighbours in V, which exists by regularity and embed the first path vertex there. Next choose a vertex x_2 in $N(x_1)$ which has at least $\frac{d}{2}|U\setminus\{x_1\}|$ neighbours in $U\setminus\{x_1\}$. In general, let X be the set of vertices $x_1,\dots,x_{i-1}$ that were already used earlier and choose a vertex x_i in $N(x_{i-1})\setminus X$ which has at least $\frac{d}{2}|(U\cup V)\setminus X|$ neighbours in $(U\cup V)\setminus X$. This is possible as long as $|U\setminus X|,|V\setminus X|\geq(2\frac{\varepsilon}{d})n$, as then we maintain that $|N(x_{i-1})\setminus X|\geq\frac{d}{2}(2\frac{\varepsilon}{d}n)=\varepsilon n$ and hence we can find such a vertex by regularity. Hence, if $d\geq\sqrt{\varepsilon}$, we get a path of length $2n(1-\sqrt{\varepsilon})$.

For the case of cycles, of course we should find even length cycles, and we can basically use the same strategy with the following slight modification. Once we embedded x_1, we choose an arbitrary set Y of size $(2\frac{\varepsilon}{d})n\leq|N(x_1)/2|$ in $N(x_1)$ that we set aside and do not use for the embedding of $x_2,x_3,\dots$. Apart from this we proceed as before, until just before the penultimate vertex. When embedding the penultimate vertex we then make sure that we choose a vertex with many neighbours in Y, so that we can then choose the last vertex in Y, completing the circle.

For binary trees, we need more care. There are different ways of doing this. Here is the outline of one: Let T be a binary tree on $2n(1-\sqrt{\varepsilon})$ vertices that has almost equally sized colour classes. This can be done for example by choosing a path and in addition adding to each inner vertex a leaf. We can then notice that we can cut up the tree into subtrees of roughly the same size: This is easy for the example tree we have just chosen, but it is also not hard to show that this can be done in general for binary trees if the sizes of the pieces are allowed to differ by a factor of 2. So assume our subtree sizes are between ξn and $2\xi n$ for some constant ξ that we choose much larger than ε but much smaller than d. For each of these subtrees we choose a natural root: We first choose a root of the whole tree, which will also be a root for one of the subtrees. We then choose the roots of the other subtrees as the vertex in the subtree closest the the tree root. So, we get constantly many subtrees, each with a root, which we will embed subtree by subtree, in a natural order given by the tree: We embed subtrees earlier if their root is closer to the tree root.

For the actual embedding we do the following: Similarly as in the cycle case, we set aside a set $Y_U \subseteq U$ and a set $Y_V \subseteq V$ of size $(10\frac{\varepsilon}{d})n$ each at the start. In these sets we will embed all the root vertices, and all other vertices will be embedded in $U \setminus Y_U$ and in $V \setminus Y_V$. We start with the first subtree, embedding its root into Y_U such that the chosen vertex has many neighbours in Y_V and also in $V \setminus Y_V$, and then embed the remaining vertices of the subtree in a search order. When in this process we embed a non-root vertex, we embed it into $V \setminus Y_V$ and $U \setminus Y_U$ proceeding analogously to the case of the path; this is possible because the subtree is small and hence neighbourhoods of chosen vertices do not get filled up. When in this process we embed a vertex that is the neighbour of a root of another subtree, we also guarantee that it is embedded to a vertex with many neighbours in the remaining vertices of Y_U or Y_V, depending on whether it is embedded to U or V. This is possible because few vertices get embedded to Y_U and Y_V. Once this subtree is embedded we can proceed to the next subtree, choosing an image for the subtree root in the neighbourhood of the already embedded neighbour of the subtree root.

The last task is harder, and a result we did not cover in the lecture (but that can be looked up) helps: The Kővári-Sós–Turán Theorem states that $2n$-vertex graphs with at least $(\frac{s-1}{2})^{1/r}(2n)^{2-1/r} = (\frac{s-1}{n})^{1/r}n^2$ edges contain a $K_{s,t}$. So if $s = r = \log n/C$, this implies that all $2n$-vertex graphs with at least $(\log n/Cn)^{C/\log n}n^2 = 2^{-C+o(1)}n^2$ edges contain a $K_{\log n/C, \log n/C}$.
On the other hand, we can consider a bipartite graph with vertex classes of size n and each edge between the vertex classes present randomly (and independently of all other choices) with probability d. It is easy to show that such a graph is an (ε, d)-regular pair. The expected number of $K_{s,s}$ in this graph is $\binom{n}{s}^2 d^{s^2} \leq n^{2s}d^{s^2}$. For $s = C\log n$ the last expression is $n^{2C\log n} \cdot d^{C^2\log^2 n} = 2^{\log^2 n(2C - C^2\log d^{-1})}$, which tends to 0 for C large enough. Hence, with high probability, this regular pair does not contain a $K_{C\log n, C\log n}$. So the largest complete bipartite graph we can hope for is of order $\Theta(\log n)$.

END OF PAPER