Colouring random 4-regular graphs

Lingsheng Shi*
lshi@math.tsinghua.edu.cn
and
Nicholas Wormald†
nwormald@uwaterloo.ca

Department of Combinatorics and Optimization
University of Waterloo
Waterloo ON
Canada N2L 3G1

Abstract

We show that a random 4-regular graph asymptotically almost surely (a.a.s.) has chromatic number 3. The proof uses an efficient algorithm which a.a.s. 3-colours a random 4-regular graph. The analysis includes use of the differential equation method, and exponential bounds on the tail of random variables associated with branching processes. A substantial part of the analysis applies to random d-regular graphs in general.

1 Introduction

We consider the uniform model $\mathcal{G}_{n,d}$ of random d-regular graphs on n labelled vertices. Many properties of random regular graphs were given in [12], and a number of results have been added since then. The chromatic number is one of the main topics in the theory of random graphs (see [5, Chapter 7] for example). Probably the latest relevant result is that of Achlioptas and Naor [2] which shows that the chromatic number of a random graph with n vertices and edge probability d/n (with d fixed) is concentrated on at most two integers, and on one value for d in some intervals.

For $\mathcal{G}_{n,d}$, the significant results on chromatic number up until now have been weak in some sense. In particular, for any fixed $d \geq 4$, the upper and lower bounds on the chromatic number χ of $G \in \mathcal{G}_{n,d}$ which were hitherto known to hold a.a.s. did not determine a single value of χ a.a.s. (We say that a property holds *asymptotically almost surely* (a.a.s.) if it holds with probability tending to 1 as $n \to \infty$. In regard to the model $\mathcal{G}_{n,d}$, we restrict nd to the even integers in all such asymptotics for feasibility reasons.) The case $d = 4$ is of particular interest since the situation for $d \leq 3$ is determined easily by combining deterministic colouring results with simple known properties of $\mathcal{G}_{n,d}$, as

*Current address: Department of Mathematics, Tsinghua University, Beijing 100084, China
†Research supported by the Canada Research Chairs Program and NSERC.

outlined below. Achlioptas and Moore [1] showed that $\mathbf{P}(\chi = 3)$ for a random graph in $\mathcal{G}_{n,4}$ is bounded below by a positive constant. In this paper, we obtain results on a class of algorithms which colour d-regular graphs for general d and obtain in particular the following "best possible" result for $d = 4$.

Theorem 1.1 *A random 4-regular graph a.a.s. has chromatic number 3.*

We also give a simple algorithm which a.a.s. colours a random 4-regular graph with three colours, based on a greedy colouring scheme (as proved in Theorem 10.1). Our analysis, expressed as it is in terms of k-colouring a random graph in $\mathcal{G}_{n,d}$ for arbitrary fixed d, lays the groundwork for a more general result which will appear in a sequel to this paper. This paper contains those portions of the analysis for which the generalisation to arbitrary d is trivial.

For completeness, we point out that random 2-regular and 3-regular graphs are a.a.s. not bipartite, as shown by the distribution of short odd cycles (see for example Bollobás [3] or [12, Section 2]). Thus they a.a.s. have chromatic number 3. This is immediate for the 2-regular case; random 3-regular graphs have no K_4 a.a.s., and so Brooks' theorem reduces the upper bound to 3.

Our starting point is an algorithm **Colour**(k) which greedily colours the vertices of a graph, using colours $1, \ldots, k$, in the following manner. At any point in the algorithm, the *type* of a vertex v is the pair (i, j) where i is the number of coloured neighbours and j is the number of colours, of $1, \ldots, k$, not available (i.e. already used on the neighbours of v). The algorithm takes as input an ordering of the possible types, as a list of priorities, and then repeatedly randomly chooses an uncoloured vertex of the highest priority type and colours it with one of its available colours chosen at random. The algorithm fails if it runs out of available colours, but in that case for convenience we demand that it proceeds using a colour from one of the neighbours (so the resulting colouring is not proper). This algorithm does succeed, on random regular graphs, if k is large enough. However, we can improve upon this slightly. What we can do is eliminate the difficulty caused near the end of the algorithm by the occurrence of short odd cycles formed by the uncoloured vertices. If all vertices in such a cycle happen to have only the same two colours available, the algorithm will fail. This type of failure prevented Achlioptas and Moore [1], who analysed a related algorithm, from being able to show that a random 4-regular graph on n vertices is a.a.s. 3-colourable.

As it turns out, a simple remedy, algorithmically, is to colour first the vertices in short cycles in the graph. Unfortunately, this simple idea complicates the analysis significantly. Let $g \geq 3$. Define a *short* cycle to be one of length at most g. By standard results, a random d-regular graph a.a.s. has at most $\log n$ short cycles, and a.a.s. no two such cycles have a vertex in common. (Also, in order to understand our argument, it may help to know that a.a.s. no vertex is adjacent to vertices from two of these short cycles, but we will not use this fact directly.) We define an algorithm **Short**(g, k) as follows. This takes as input a d-regular graph G. As above, it is understood that the input also includes a priority list of types.

Algorithm **Short**(g, k)

1 properly 3-colour the vertices in the short cycles of G;
2 apply **Colour**(k).

We say this algorithm (and other colouring algorithms involving k colours) has *failed* at any time that a vertex has k differently coloured neighbours. It *succeeds* if, up until the time that all vertices of G have been coloured, it has not failed.

We claim that, for $d = 4$, the algorithm **Short**$(g, 3)$ (with a suitable list of priorities) succeeds with probability at least $1-\epsilon_g+o(1)$ when applied to a random 4-regular graph on n vertices, as $n \to \infty$. Here ϵ_g is a function which tends to 0 as $g \to \infty$. Theorem 1.1 is a consequence of this, using the following simple result.

Fact If for all fixed $g > 0$, $f(n, g) = o(1)$ as $n \to \infty$, then for some function $g = g(n)$ going to infinity (sufficiently slowly), $f(n, g) = o(1)$.

This paper continues the approach of the second author, of analysing algorithms on random structures using a differential equation method. The version of this method we use for greedy algorithms was introduced in [10]. Extensions of the technique and of related ones will be used in the sequel [7] to obtain some similar results for the chromatic number of random d-regular graphs when d is small. Many of the results in the present paper will be used for this.

In the next section we give more detail on some of the delicate parts of the argument, intermingled with some useful definitions. Section 3 introduces a basic colouring process on a model of random d-regular graphs. Section 4 studies the basic combinatorics of the situation in order to derive the differential equations which explain the behaviour of the colouring algorithms under consideration. Part of this discussion is nonrigorous but it provides an intuitive understanding of the processes under consideration, and additionally guides the later rigorous development.

We call a type (i, j) *dangerous* if $j \geq k - 1$ and $i < d$. In order to ensure that the greedy algorithm does not colour itself into a corner, we need to know that it can usually succeed with colouring dangerous types of vertices. These vertices are not nice to have around, because they are uncoloured vertices with at most one colour not already used on their neighbours. Colouring a vertex with a dangerous neighbour can leave that neighbour with no available colours, so allowing dangerous vertices to proliferate would cause problems. To show that the numbers of vertices of dangerous type do not build up (with a suitable priority list), we use a branching process argument, the foundations of which are in Section 5. This is then applied in Section 6 to achieve the desired result.

Section 7 shows that the process will usually get off to a good start without any problems. However, it is hard to push the analysis of the random process to the very end using differential equations. Instead we can stop when the remaining graph is sparse enough, in a certain sense. Section 8.1 defines such a time, t_1, and the rest of Section 8 shows that in the typical stages of the colouring process up to time t_1, the lack of short cycles ensures that the vertices of dangerous type, which definitely can occur often during the process, have low probability of being able to act in collusion (which could cause a real problem). The probability of a problem is bounded as a function of the (fixed) girth g, and this function is shown to go to 0 as g goes to infinity. Section 9

shows that the likelihood of having a problem after t_1 is reached is negligible. Section 10 then amalgamates the conclusions of the earlier sections, and these results are finally used in Section 11, for the case $d = 4$, to prove Theorem 1.1.

2 Definitions and outline

To analyse the performance of the algorithm **Short**(g, k) on $G \in \mathcal{G}_{n,d}$, we use the usual *pairing* model of random regular graphs. A pairing is a perfect matching of $dn/2$ pairs with dn points which are partitioned into n cells, each containing exactly d points. Such a pairing P corresponds to a multigraph (with loops and multiple edges) $G(P)$ in which the cells are regarded as vertices and the pairs are edges; we often speak of the cells as vertices. As used in [11] and many other places, a uniformly random pairing can be generated by repeatedly choosing (using any rule whatsoever) a point x to be paired, and then randomly choosing the point y to be paired with x from the remaining unpaired points. We refer to this choice of y as *exposing* the pair containing x.

We define a process **PColour**(k), corresponding to **Colour**(k), as follows. It both generates a pairing P and colours the vertices of $G(P)$. At each step, a vertex v of $G(P)$ is selected and coloured using the same rule as **Colour**(k). (If applied to the empty pairing, it chooses any vertex and colours it any colour.) Then the remaining points in v are paired consecutively, each time choosing a random unpaired point to be paired with. (A more detailed definition is given in Section 3.) By the observation above, if this begins with the empty pairing, the result is a uniformly generated random pairing.

Let $\mathcal{S}$ be a specified set of pairs of a potential pairing which induce disjoint cycles in $G(P)$. It should cause no confusion to regard $\mathcal{S}$ alternatively as a set of edges, or as the subgraph of $G(P)$ it induces, and we permit ourselves a similar latitude in speaking about pairs in the pairing in other respects as edges of the corresponding graph. We define a process **PComplete**$(k, \mathcal{S})$ as follows. It both generates a pairing P and colours the vertices of $G(P)$.

Process **PComplete**$(k, \mathcal{S})$

1 Begin with dn points and only the pairs in $\mathcal{S}$;
2 Properly 3-colour the vertices containing points in $\mathcal{S}$ and expose the remaining pairs containing their points;
3 Apply **PColour**(k) to this initial pairing to complete the generation of a coloured pairing.

Roughly speaking (with modifications to be explained below), we will use this process to analyse the algorithm **Short** applied to a random d-regular graph. To do this, we use it to analyse a colouring of a random graph G_0 such that $G_0 \cup \mathcal{S}$ is a d-regular graph on n vertices, with no short cycles other than the cycles in $\mathcal{S}$ (with uniform probability distribution subject to this constraint). The process **PComplete**$(k, \mathcal{S})$ itself does not ensure that the cycles $\mathcal{S}$ are exactly the short cycles of the graph $G(P)$ which results; this must be accounted for in the analysis. An *unwanted* cycle is a cycle created during

the execution of this process which is short but not contained in $\mathcal{S}$. (In practice, loops, or cycles of length 1, are always unwanted.)

An edge, or the corresponding pair, is called *free* if it is incident with at least one uncoloured vertex. Let t_1 denote a time at which the graph induced by the free edges is sparse (in a sense to be made precise). For certain d and k we will show (Lemma 8.2) that, for a certain time $t_1(d,k)$, Process **PComplete**$(k,\mathcal{S})$ does not fail before time t_1 (i.e. creates no vertex with k differently coloured neighbours until that time) with probability at least $1-\epsilon_g+o(1)$ where $\epsilon_g \to 0$ as $g \to \infty$, conditional upon there being no unwanted cycles up to time t_1. We also show (Lemma 9.1) that, with probability at least $1-\epsilon_g+o(1)$, given that the graph at time t_1 has no unwanted cycles, the subgraph induced by the free edges, conditional upon no unwanted cycles, is going to be a forest. We also show that, at a point near t_1, every vertex has at least two colours available. Since a forest is 2-chooseable, the rest of the graph can then be coloured (and the process then succeeds, provided the vertices with only one colour available have priority over others). Finally, recall that G_0 is a random graph with a *fixed* set of short cycles. We will show how to deduce from this the conclusion we want about the algorithm **Short**(g,k) provided that the algorithm on G_0 succeeds with probability at least $1-\epsilon_g$, for any fixed $\mathcal{S}$, where the function $\epsilon_g \to 0$ as $g \to \infty$. For any values of k and d that this plan succeeds, it implies that $G \in \mathcal{G}_{n,d}$ is a.a.s. k-colourable. To prove Theorem 1.1 only requires verifying that this is the case for $d=4$ and $k=3$.

Lemma 8.2 is the heart of the combinatorial part of our proof, so it is worth sketching in more detail how its proof goes. Considering the process **PComplete**$(k,\mathcal{S})$, define a *bad* edge (or pair) to be an edge (or the corresponding pair) which does not lie in a short cycle and, when it is added during the process, joins two vertices which each have at least $k-1$ different colours on their neighbours (before the edge is added). Regarding the pairing process, a bad pair is a pair corresponding to a bad edge. If there are no bad pairs then the colouring clearly succeeds, conditioned upon no unwanted cycles. We show that the probability of a bad edge occurring before time t_1 is small, and also that it is not altered much without the extra conditioning on no unwanted cycles occurring. (The fact that this is "more true" for larger g is very important.)

For reasons as explained in Section 1, it is crucial that we select t_1 so that the algorithm has not before t_1, typically, begun creating large numbers of vertices of dangerous type. We proceed as follows at each step in the algorithm, we compute the expected changes of the numbers of vertices with all different types, and use a system of differential equations to approximate the process. We add new variables for the differential equation which keep track of the number of paths of all possible types of length less than g which have been created. All these variables will be sharply concentrated, and given these variables we know, at every step, the asymptotic probability that a cycle of given length is created, and also a bound on the probability of a bad edge. These bounds work even when conditioning on no unwanted cycles having occurred so far in the process, because that event has probability bounded away from 0, and yet the variables are sharply concentrated. In this way, we have that a random pairing after t_1 steps, conditioned on the event that $\mathcal{S}$ is the set of short cycles, expects a very small number of bad edges, as required for Lemma 8.2.

3 The process PColour(k)

In this section, we describe in detail the process **PColour**(k), which is the pairing model version of a procedure which simultaneously generates a random d-regular graph and colours it with k colours.

Given a partial colouring of a partial pairing P — i.e., some of the vertices (cells) have been coloured — we classify the uncoloured cells according to the number of coloured neighbours: an uncoloured cell is of *type* (i, j), if it has i points which are paired and exactly j colours which appear in its current neighbours. All types of vertices will be assigned the priorities 0,1,2,... (a larger number denoting higher priority) according to a function `prio`(i, j) (integer i and j). The randomised greedy process **PColour**(k), which properly k-colours the vertices of pairings, is as follows, taking as input a predetermined function `prio` and a partially coloured partial pairing.

while there is at least one uncoloured vertex **do**
  part (i):
    let (i, j) denote the highest priority type of an uncoloured vertex;
    select u.a.r. a vertex v of type (i, j);
  part (ii):
    colour v with a colour chosen u.a.r. from the colours not appearing
    on neighbours of v;
    expose all pairs which contain a point in v;
end while

By the comments on random pairings in the previous section, the pairings produced by this algorithm occur uniformly at random (conditional upon containing the initial partial pairing). The repeated step is broken into two parts for later reference.

Note that the priority of pairs (i, j) with $i = d$ is immaterial since colouring vertices of such type cannot affect the remainder of the algorithm. So for simplicity we assume that all such vertices have negative priority, and only those with $i < d$ and $j < k$ need to be specified.

4 Derivation of differential equations for the algorithm

In order to analyse the performance of the randomised algorithm, we use a system of differential equations to approximate the expected changes in the variables describing the state of the algorithm during its execution. An exposition of this method can be found in [11], which includes various examples of graph-theoretic optimisation problems.

The analysis of the process **PColour**(k) we give here is similar to the preliminary examination of the bisection width algorithm in [4]. Again, it is not fully rigorous but prepares the way for the later rigorous arguments by supplying most of the explanation for the relevance of the main differerential equations.

Informally speaking, in a typical part of the algorithm, there will be vertices of one particular type, (i,j), which are plentiful in the pairing but are quite regularly chosen as v in the algorithm. Vertices of types with higher priority may also be regularly chosen, but will be rare and regularly be used up entirely (at which point another vertex of type (i,j) again will be coloured.) In this situation, we say that (i,j) is the *basic* type. The algorithm will typically pass through *phases*, determined by time at which, roughly speaking, the basic type changes. A phase finishes when either vertices of higher priorities than the current basic type become plentiful, or those with the current basic type become very scarce. The boundaries of the phases are best defined precisely in terms of the solution of a system of differential equations which we proceed to derive now. Luckily, in the case $d = 4$ there will only be one phase, and we can postpone coming fully to grips with this issue until the sequel [7].

We call each iteration of the **while** loop of the process **PColour**(k) a *step*. An uncoloured cell is said to be of type (i,C), if it has i points which are paired and $C \subseteq \{1,\dots,k\}$ is the set of colours which appear in its neighbours. There should be no ambiguity with type (i,j), already defined, since j is a number and C is a set of colours.

At the start of some step, let $Y_{i,C}$ denote the number of uncoloured vertices of type (i,C), and let S denote the number of points not exposed in vertices yet. Then

$$S = \sum_{0\le i<d} \sum_{C\subseteq\{1,\dots,k\}} (d-i)Y_{i,C}. \tag{4.1}$$

We may assume for this discussion that $S > 0$.

Suppose that a vertex v has just been coloured c by the greedy algorithm and an unpaired point in v is being paired with a randomly chosen point, x. Let $r^c_{i,C}$ denote the expected contribution to $\Delta Y_{i,C}$, the change of $Y_{i,C}$, due to the change in status of the vertex u containing x. The probability that u is of type (i,C) is $(d-i)Y_{i,C}/S$ (except for a correction of size $O(1/S)$ due to the change in status of v). Hence, ignoring terms of size $O(1/S)$,

$$r^c_{i,C} = \begin{cases} (d-i+1)(Y_{i-1,C\setminus\{c\}} + Y_{i-1,C})/S - (d-i)Y_{i,C}/S & \text{if } c\in C \\ -(d-i)Y_{i,C}/S & \text{if } c\notin C. \end{cases}$$

We will continue to ignore terms of size $O(1/S)$ in the estimates in this section.

Now let $r^{i',C'}_{i,C}$ denote the expected contribution to $\Delta Y_{i,C}$, the change in $Y_{i,C}$, due to exposing the pair containing the point x of the vertex u of type (i',C'). Noting that for each colour $c\notin C'$, the probability that the vertex v was coloured c is exactly $1/(k-|C'|)$, we obtain

$$r^{i',C'}_{i,C} = \frac{1}{k-|C'|} \sum_{c\in\{1,\dots,k\}\setminus C'} r^c_{i,C} = \beta^{i',C'}_{i,C} - (d-i)Y_{i,C}/S \tag{4.2}$$

where

$$\beta^{i',C'}_{i,C} = \frac{1}{k-|C'|} \sum_{c\in C\setminus C'} (d-i+1)(Y_{i-1,C\setminus\{c\}} + Y_{i-1,C})/S. \tag{4.3}$$

Thus the expected change in $Y_{i,C}$ given that *one vertex* v of type (i', C') is coloured is

$$\mathbf{E}[\Delta Y_{i,C} \mid v \text{ is of type } (i', C')] = (d - i')r_{i,C}^{i',C'} - \delta_{(i,C)=(i',C')}, \tag{4.4}$$

where δ_A is 1 if A is true and 0 otherwise.

At this point we begin to introduce nonrigorous components into the argument. We assume (justified, for some applications, later) that the probability that a vertex of type (i, C) is selected to be coloured basically depends on the cardinality of C and not what the particular colours are. With this in mind, we loosely define $\phi_{i,j}$ to be the probability of processing a vertex of type (i, C) for each $|C| = j$ at some given step. (This use of "probability" does not make sense in a rigorous discussion since calculations such as above and below depend on the values of the $Y_{i,C}$, and due to prioritisation, these influence the type (i, C) to be chosen at a given step. However, $\phi_{i,j}$ can be interpreted as the proportion of steps which process such a vertex over some period of time which, although long, is nevertheless insignificant compared to n. In a later section, this discussion will be revised and ϕ will be endowed with a rigorous interpretation.)

Let (i_0, j_0) be the basic type at some point in the algorithm. Assuming $\sum_{|C|=j_0} Y_{i_0,C}$ is "large", no vertex of lower priority will be chosen as v and coloured, for many more steps of the algorithm. Let B' denote the set of types (i, j) of vertices with higher priority than the basic, (i_0, j_0), and let $B = B' \cup \{(i_0, j_0)\}$, i.e. the set of all types of basic priority and higher. Note that in a large number of consecutive steps, the prioritisation ensures that the vertices of high priority are created at the same rate as being destroyed. This lets us estimate the values of the ϕ's. By computing the expected changes of the numbers of vertices with high priorities, and also setting $Y_{i,C} = Y_{i,D}$ if $|C| = |D|$ (which will in some sense be justified in the later rigorous analysis, but for now it suffices to guess that these variables will usually be approximately equal), we obtain after a number of algebraic manipulations (which we do not provide here, as similar equations will arise from the rigorous discussion in Section 6)

$$\phi_{i,j} = \tilde{\zeta} r_{i,j}, \ (i, j) \in B', \tag{4.5}$$

$$\phi_{i_0,j_0} = \binom{k}{j_0}^{-1} \left(1 - \sum_{(i,j)\in B'} \phi_{i,j} \binom{k}{j}\right), \tag{4.6}$$

where

$$r_{i,j} = (d - i + 1)j(\overline{Y}_{i-1,j} + \overline{Y}_{i-1,j-1})/(kS) - (d - i)\overline{Y}_{i,j}/S, \tag{4.7}$$

$$\overline{Y}_{i,j} = \binom{k}{j}^{-1} \sum_{|C|=j} Y_{i,C}, \tag{4.8}$$

$$\tilde{\zeta} = (d - i_0)/T, \tag{4.9}$$

$$T = 1 + \sum_{(i,j)\in B'} (i - i_0)\binom{k}{j} r_{i,j}. \tag{4.10}$$

In this computation, $\tilde{\zeta}$ represents the expected number of points in a vertex processed.

Assuming the validity of these equations and noting (4.4), the expected change $\Delta Y_{i,C}$ in the random variable $Y_{i,C}$ in one step of **PColour**(k) is given by

$$\mathbf{E}(\Delta Y_{i,C}) = \sum_{(i',|C'|)\in B} (d-i') r_{i,C}^{i',C'} \phi_{i',|C'|} - \phi_{i,|C|}. \tag{4.11}$$

The final subtracted term is due to the change in type of the vertex being coloured.

As usual [11], the above expected changes suggest a system of differential equations whose solution describes the likely approximate behaviour of the variables $Y_{i,C}$ (again, formal justification is in later sections). In the equations, each $\mathbf{E}(\Delta Y_{i,C})$ appears as the derivative $Y'_{i,C}$ (as functions of the number t of iterations). It is convenient to scale both time and the variables by a factor $1/n$, approximating $Y_{i,C}/n$ by $y_{i,C}$ and t/n by x. Since the scaling is by the same factor in both cases, each $\mathbf{E}(\Delta Y_{i,C})$ approximates the derivative $y'_{i,C}$. Let $\mathbf{y}$ denote the vector $(y_{i,C})$, $0 \le i < d$, $C \subseteq \{1,...,k\}$, and replace S/n by $s(\mathbf{y})$, $\overline{Y}_{i,j}/n$ by $\overline{y}_{i,j}(\mathbf{y})$, T by $a(\mathbf{y})$, $r_{i,j}$ by $b_{i,j}(\mathbf{y})$, $r_{i,C}^{i',C'}$ by $b_{i,C}^{i',C'}(\mathbf{y})$, $\tilde{\zeta}$ by $\zeta(\mathbf{y})$, and ϕ by θ. Then (4.11) becomes

$$y'_{i,C} = \sum_{(i',|C'|)\in B} (d-i') b_{i,C}^{i',C'}(\mathbf{y}) \theta_{i',|C'|}(\mathbf{y}) - \theta_{i,|C|}(\mathbf{y}).$$

where (4.1), (4.8), (4.7), (4.2), (4.10), (4.9), (4.5), and (4.6) give

$$\begin{aligned}
s(\mathbf{y}) &= \sum_{0\le i<d;\ C\subseteq\{1,...,k\}} (d-i) y_{i,C}, \\
\overline{y}_{i,j}(\mathbf{y}) &= \binom{k}{j}^{-1} \sum_{|C|=j} y_{i,C}, \\
b_{i,j}(\mathbf{y}) &= (d-i+1)j(\overline{y}_{i-1,j} + \overline{y}_{i-1,j-1})/(ks(\mathbf{y})) - (d-i)\overline{y}_{i,j}/s(\mathbf{y}), \\
b_{i,C}^{i',C'}(\mathbf{y}) &= \frac{1}{k-j'} \sum_{c\in C\setminus C'} \frac{(d-i+1)(y_{i-1,C\setminus\{c\}} + y_{i-1,C}) - (d-i)y_{i,C}}{s(\mathbf{y})}, \\
a(\mathbf{y}) &= 1 + \sum_{(i,j)\in B'} (i-i_0) \binom{k}{j} b_{i,j}(\mathbf{y}), \\
\zeta(\mathbf{y}) &= (d-i_0)/a(\mathbf{y}), \\
\theta_{i,j}(\mathbf{y}) &= \begin{cases} 0 & \text{if } (i,j)\notin B \\ \zeta(\mathbf{y}) b_{i,j}(\mathbf{y}) & \text{if } (i,j)\in B', \end{cases} \\
\theta_{i_0,j_0}(\mathbf{y}) &= \binom{k}{j_0}^{-1} \left(1 - \sum_{(i,j)\in B'} \binom{k}{j} \zeta(\mathbf{y}) b_{i,j}(\mathbf{y})\right).
\end{aligned} \tag{4.12}$$

The initial condition is $y_{0,\emptyset}(0) = 1$, and $y_{i,C}(0) = 0$ otherwise.

A system of first order differential equations with given initial values has a unique solution on any bounded domain in which the derivative functions satisfy a Lipschitz condition (as these ones do if $s(\mathbf{y})$ is bounded away from 0). By the symmetry of the equations and the initial conditions, upon setting $y_{i,C} = y_{i,j}$ for all C with $|C| = j$, and

similarly for the other variables, we obtain a set of equations with a reduced number of variables that determines the solutions. So we henceforth write $y_{i,j}$ for $y_{i,C}$ when $|C| = j$. We now have

$$\begin{aligned}\sum_{|C'|=j'} b_{i,C}^{i',C'}(\mathbf{y}) &= \sum_{|C'|=j'} \frac{1}{k-j'} \sum_{c\in C\setminus C'} \frac{(d-i+1)(y_{i-1,C\setminus\{c\}} + y_{i-1,C}) - (d-i)y_{i,C}}{s(\mathbf{y})} \\ &= \binom{k}{j'} \left(\frac{(d-i+1)|C|(y_{i-1,|C|-1} + y_{i-1,|C|})}{ks(\mathbf{y})} - \frac{(d-i)y_{i,|C|}}{s(\mathbf{y})} \right) \\ &= \binom{k}{j'} b_{i,|C|}(\mathbf{y}),\end{aligned}$$

where the second equality follows from the fact that if the $k-j'$ elements of the complement of C' are randomly chosen, the probability that the i'th element is in C is $|C|/k$, and so by linearity of expectation

$$\sum_{|C'|=j'} \sum_{c\in C\setminus C'} 1 = \sum_{|C'|=j'} |C \setminus C'| = \binom{k}{j'}(k-j')|C|/k.$$

We may also simplify by noting that

$$\begin{aligned}\sum_{(i,j)\in B} (d-i)\binom{k}{j}\theta_{i,j}(\mathbf{y}) &= (d-i_0)\left(1 - \sum_{(i,j)\in B'} \binom{k}{j}\zeta(\mathbf{y})b_{i,j}(\mathbf{y})\right) \\ &\quad + \sum_{(i,j)\in B'} (d-i)\binom{k}{j}\zeta(\mathbf{y})b_{i,j}(\mathbf{y}) \\ &= d - i_0 + \zeta(\mathbf{y}) \sum_{(i,j)\in B'} (i_0 - i)\binom{k}{j} b_{i,j}(\mathbf{y}) \\ &= \zeta(\mathbf{y})a(\mathbf{y}) + \zeta(\mathbf{y}) \sum_{(i,j)\in B'} (i_0 - i)\binom{k}{j} b_{i,j}(\mathbf{y}) \\ &= \zeta(\mathbf{y}).\end{aligned}$$

With slight abuse of terminology, we still let $\mathbf{y}$ denote the vector $(y_{i,j})$, $0 \le i < d$ and $0 \le j < k$. The simplified equations then become

$$y'_{i,j} = \zeta(\mathbf{y})b_{i,j}(\mathbf{y}) - \theta_{i,j}(\mathbf{y}), \tag{4.13}$$

where

$$\begin{aligned}
s(\mathbf{y}) &= \sum_{0\le i<d;\ 0\le j<k} (d-i)\binom{k}{j} y_{i,j},\\
b_{i,j}(\mathbf{y}) &= (d-i+1)j(y_{i-1,j}+y_{i-1,j-1})/\big(ks(\mathbf{y})\big) - (d-i)y_{i,j}/s(\mathbf{y}), \qquad (4.14)\\
a(\mathbf{y}) &= 1+\sum_{(i,j)\in B'} (i-i_0)\binom{k}{j} b_{i,j}(\mathbf{y}), \qquad (4.15)\\
\zeta(\mathbf{y}) &= (d-i_0)/a(\mathbf{y}), \qquad (4.16)\\
\theta_{i,j}(\mathbf{y}) &= \begin{cases} 0 & \text{if } (i,j)\notin B\\ \zeta(\mathbf{y})b_{i,j}(\mathbf{y}) & \text{if } (i,j)\in B'\\ \left(1-\sum_{(i',j')\in B'}\binom{k}{j'}\zeta(\mathbf{y})b_{i',j'}(\mathbf{y})\right)\binom{k}{j}^{-1} & \text{if } (i,j)=(i_0,j_0). \end{cases} \qquad (4.17)
\end{aligned}$$

In order to further simplify the above differential equations, change variables from x to $z=(d-s)/2$. Note that for each edge added, nz increases by 1. So

$$\frac{\mathrm{d}z}{\mathrm{d}x} = -\frac{1}{2}\frac{\mathrm{d}s}{\mathrm{d}x}, \qquad (4.18)$$

and clearly $z(0)=0$. As noted earlier, ζ represents the expected number of points in a vertex when the colouring operation is performed, i.e. the expected number of edges added per vertex processed. So we also write

$$\frac{\mathrm{d}z}{\mathrm{d}x} = \zeta(\mathbf{y}).$$

Hence

$$\begin{aligned}
\frac{\mathrm{d}y_{i,j}}{\mathrm{d}z} &= \frac{\mathrm{d}x}{\mathrm{d}z}y'_{i,j} = b_{i,j}(\mathbf{y}) - \theta_{i,j}(\mathbf{y})/\zeta(\mathbf{y})\\
&= \begin{cases} b_{i,j}(\mathbf{y}) & \text{if } (i,j)\notin B\\ 0 & \text{if } (i,j)\in B'\\ b_{i,j}(\mathbf{y}) - \left(\dfrac{a(\mathbf{y})}{d-i_0} - \displaystyle\sum_{(i',j')\in B'}\binom{k}{j'}b_{i',j'}(\mathbf{y})\right)\binom{k}{j}^{-1} & \text{if } (i,j)=(i_0,j_0) \end{cases} \qquad (4.19)
\end{aligned}$$

where, in terms of z,

$$b_{i,j}(\mathbf{y}) = \frac{(d-i+1)j}{(d-2z)k}(y_{i-1,j}+y_{i-1,j-1}) - \frac{d-i}{d-2z}y_{i,j}.$$

To simplify further we use the following lemma, which will also be useful later.

Lemma 4.1 *For $T>0$ and $a(\mathbf{y})>0$ we have*

$$\begin{aligned}
\phi_{i_0,j_0} &= \left(1-\sum_{(i,j)\in B'}(d-i)r_{i,j}\binom{k}{j}\right)\left(T\binom{k}{j_0}\right)^{-1},\\
\theta_{i_0,j_0} &= \left(1-\sum_{(i,j)\in B'}(d-i)b_{i,j}\binom{k}{j}\right)\left(a\binom{k}{j_0}\right)^{-1}.
\end{aligned}$$

Proof. We show the equality for ϕ. A similar argument applies for θ.

$$\begin{aligned}
\phi_{i_0,j_0}\binom{k}{j_0} &= 1 - \sum_{(i,j)\in B'} \phi_{i,j}\binom{k}{j} \text{ by (4.6)} \\
&= 1 - \sum_{(i,j)\in B'} \zeta r_{i,j}\binom{k}{j} \text{ by (4.5)} \\
&= 1 - \sum_{(i,j)\in B'} (d-i_0) r_{i,j}\binom{k}{j} T^{-1} \text{ by (4.9)} \\
&= \left(T - \sum_{(i,j)\in B'} (d-i_0) r_{i,j}\binom{k}{j}\right) T^{-1} \\
&= \left(1 - \sum_{(i,j)\in B'} (d-i) r_{i,j}\binom{k}{j}\right) T^{-1} \text{ by (4.10).} \quad \blacksquare
\end{aligned}$$

With this lemma, (4.19) becomes

$$\frac{\mathrm{d}y_{i,j}}{\mathrm{d}z} = \begin{cases} b_{i,j}(\mathbf{y}) & \text{if } (i,j) \notin B \\ 0 & \text{if } (i,j) \in B' \\ b_{i,j}(\mathbf{y}) + \dfrac{\sum_{(i',j')\in B'} (d-i')\binom{k}{j'} b_{i',j'}(\mathbf{y}) - 1}{(d-i)\binom{k}{j}} & \text{if } (i,j) = (i_0,j_0). \end{cases} \tag{4.20}$$

Recall that $s = d - 2z$. Then every derivative is just a linear combination of the ratios $u_{i,j} = y_{i,j}/s$, except that in the case of (i,j) basic, there is an extra constant term.

Note that $\mathbf{b}$ (the vector of the $b_{i,j}$) is of the form $B\mathbf{y}$ where the rows of the matrix B are indexed by (i,j), as are the columns. Hence the equations are of the form

$$\mathbf{y}' = B\mathbf{y}/s + \mathbf{c}$$

where $s = d - 2z$, B is a matrix and $\mathbf{c}$ is a constant vector with

$$c_{(i,j)} = \begin{cases} -\left((d-i)\binom{k}{j}\right)^{-1} & \text{if } (i,j) = (i_0,j_0), \\ 0 & \text{otherwise.} \end{cases}$$

Now put

$$u_{i,j} = y_{i,j}/s, \qquad s\frac{\mathrm{d}u_{i,j}}{\mathrm{d}z} = y'_{i,j} - u_{i,j}s'$$

(differentiation with respect to z). Then $s' = -2$, and $s = \frac{\mathrm{d}z}{\mathrm{d}v}$ is solved to find $v = -\frac{1}{2}\log(d-2z)$ (any solution will do). Hence the left side of the equation above becomes $\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}v}$, and letting $A = B + 2I$ where I is the identity matrix, i.e. we have

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}v} = A\mathbf{u} + \mathbf{c}, \tag{4.21}$$

a linear system of first order differential equations, which can be written explicitly as

$$\frac{\mathrm{d}u_{i,j}}{\mathrm{d}v} = \begin{cases} b_{i,j}(\mathbf{u}) + 2u_{i,j} & \text{if } (i,j) \notin B \\ 0 & \text{if } (i,j) \in B' \\ b_{i,j}(\mathbf{u}) + 2u_{i,j} + \dfrac{-k + \sum_{(i',j') \in B'} (d-i')\binom{k}{j'} b_{i',j'}(\mathbf{u})}{k(d-i)\binom{k}{j}} & \text{if } (i,j) = (i_0, j_0), \end{cases} \tag{4.22}$$

where

$$b_{i,j}(\mathbf{u}) = (d-i+1)j(u_{i-1,j} + u_{i-1,j-1})/k - (d-i)u_{i,j}.$$

At this point we may describe (without justification) what we expect to happen in the case of applying **PColour(k)** beginning with the empty pairing, in the case $d = 4$. The numbers of vertices with two coloured neighbours will initially remain small. At least for $d = 4$, the process will proceed with the variables approximately following the trajectories suggested by these differential equations. After some time the uncoloured part of the graph will be so sparse that it can be coloured easily using three colours. Thus, if there is no problem of a vertex requiring four colours during the earlier part of the process, a 3-coloured random 4-regular graph will be produced. Unfortunately, there will occasionally be vertices produced in the process which have three coloured neighbours, and if two of them are adjacent, there can be trouble. This will happen more often if the graph has short cycles. This is why we use **PComplete**$(k, \mathcal{S})$.

To be able to interpret the solution of the differential equations as a reasonably accurate representation of the greedy colouring process, we need to introduce either deprioritisation or (as in some of the earlier papers) branching processes. For the present argument, we need branching processes in any case to ensure that the numbers of vertices of non-basic type remain extremely small (a.a.s.), and deprioritisation is not required. So we deal with branching processes next.

5 Aside: general branching process results

We say that a nonnegative integer valued random variable X has a K, c *tail* if for all nonnegative integers s

$$\mathbf{P}(X \geq s) \leq Ke^{-cs}.$$

If X has such a tail for some $K, c > 0$, we say that X has an *exponentially small tail*. Crámer essentially proved the following (see [8]), where we call the *size* of a branching process the random number of individuals born in the process before it extinguishes.

Theorem 5.1 *Let K, c be positive reals. Let Z be a nonnegative integer valued random variable with a K, c tail and with $\mathbf{E}[Z] = \mu < 1$. Let T be the size of the Galton-Watson branching process in which each node, independently, has Z children. Then there exist positive K^+, c^+, dependent only on μ, K and c, such that T has a K^+, c^+ tail.*

We consider multi-type branching processes, in which the individuals have different types but those of each type have the same characteristics. For the algorithms in this paper, the branching processes we need to look at have a particularly simple form.

Let us define a *regular multi-type branching process* to be a branching process with k types of individuals, each individual u giving birth to some number of individuals of each or any of the various types, with the following properties.

(i) Letting $B(u)$ denote the random vector $(X_1(u), \ldots, X_k(u))$ where $X_i(u)$ is the number of children of u of type i, the random vectors $B(u)$ and $B(v)$ are independent for different individuals u and v, and identically distributed for u and v of the same type.

(ii) Suppose that the number of type j children of a vertex of type i is distributed as Z_{ij} (so this has the marginal distribution of $X_j(u)$ for u of type i). There exist nonnegative reals α_i and β_i, $1 \le i \le k$, such that for all $1 \le i \le k$ and $1 \le j \le k$,

$$\mathbf{E}Z_{ij} = \alpha_i \beta_j.$$

There is no need that $X_j(u)$ and $X_i(u)$ be independent for a particular u, and in fact we consider regular processes where dependence holds.

Note that the rank of the matrix $(\alpha_i \beta_j)$ is 1 and so it has $k-1$ eigenvalues equal to 0 and largest eigenvalue equal to $\sum_{i=1}^k \alpha_i \beta_i$. The standard theory of multitype branching processes gives that the process dies out a.a.s. if this largest eigenvalue is less than 1. But we need more, as given in the following lemma.

Lemma 5.2 *Suppose that, for a regular multi-type branching process $\mathcal{P}$ as defined above,*

$$\sum_{i=1}^k \alpha_i \beta_i < 1 \tag{5.1}$$

and that for all i and j, Z_{ij} has an exponentially small tail. Let $\mathcal{P}$ begin with one individual. Then the size of $\mathcal{P}$ has an exponentially small tail.

Proof. For s fixed, $1 \le s \le k$, consider the process $\mathcal{P}(s)$ induced from the definition of $\mathcal{P}$ by ignoring individuals of all types $i > s$. This is defined for s types of individual, and when u has type i, $\mathbf{E}[X_j(u)] = \alpha_i \beta_j$ for j in $\{1, \ldots, s\}$. We show by induction on s that the size of $\mathcal{P}(s)$ has an exponentially small tail.

First, $\mathcal{P}(1)$ is a Galton-Watson branching process with $\mathbf{E}[X_1(u)] = \alpha_1 \beta_1 < 1$ for u of type 1, and the $X_1(u)$ has a K, c component tail by assumption. So by Theorem 5.1, $\mathcal{P}(1)$ has an exponentially small tail.

For the inductive step, consider $\mathcal{P}(s)$ for $s > 1$. For u of type s, let $Q(u)$ denote all descendents of u which are of type strictly less than s and have no ancestor of type s which is a descendent of u. Then $Q(u)$ is determined by a copy of $\mathcal{P}(s-1)$ which begins with the children of u of type less than s. So by the inductive hypothesis, $|Q(u)|$ has an exponentially small tail. Now consider the set $Q'(u)$ of individuals in $\mathcal{P}(s)$ which are s-type children of individuals in $Q(u) \cup \{u\}$. Since each Z_{is} has an exponentially small tail, it follows that $|Q'(u)|$, being the sum of $|Q(u)|+1$ variables of the form Z_{is}, also has an exponentially small tail. (There are several ways to see this. One way would be to consider the probability generating functions $f(x)$ and $g(x)$ for $|Q(u)|+1$ and a variable

W that (just) stochastically dominates all of the Z_{is}. Since they have exponentially small tails, so does W. Equivalently, the radius of convergence of $g(x)$ is strictly greater than 1. So is that of $f(x)$. Since $g(1) = 1$, the composition $f(g(x))$ also has radius of convergence strictly greater than 1.)

In $\mathcal{P}(s)$ there is an embedded Galton-Watson process $\mathcal{P}^-(s)$, for which the individuals are the ones in $\mathcal{P}$ of type s, and the children of an individual u in $\mathcal{P}(s)$ are the individuals in $Q'(u)$. We claim that $\mathbf{E}[|Q'(u)|] < 1$ (proof below). It then follows by Theorem 5.1 and the observation above that the size of $\mathcal{P}^-(s)$ has an exponentially small tail. The lemma follows from this statement and the fact that the size of $\mathcal{P}(s)$ is equal to the sum of $|Q(u)| + 1$ over all individuals u in $\mathcal{P}^-(s)$. This then has an exponentially small tail, for the same reason as argued for $|Q'(u)|$ above.

The proof is finished by showing that $\mathbf{E}[|Q'(u)|] < 1$. We can generate $Q(u) \cup Q'(u)$ by running a branching process $\mathcal{P}'(s)$ which is like $\mathcal{P}(s)$, and begins with one s-type individual u, but altering the rules so that no s-type individuals other than u may give birth. Let r_i be the expected number of individuals of type i actually born (not counting u) in $\mathcal{P}'(s)$. Then

$$r_i = \sum_{j=1}^{s} r_j^* \alpha_j \beta_i \tag{5.2}$$

for all i, where $r_i^* = r_i$ for $i < s$ and $r_s^* = 1$ (standing for u). Put

$$m = \sum_{j=1}^{s} r_j^* \alpha_j$$

Then from (5.2),

$$r_i = m\beta_i, \quad 1 \le i \le s. \tag{5.3}$$

Thus

$$\begin{aligned} m &= \alpha_s + \sum_{j=1}^{s-1} r_j \alpha_j \\ &= \alpha_s + \sum_{j=1}^{s-1} m\beta_j \alpha_j \\ &< \alpha_s + m(1 - \alpha_s \beta_s) \end{aligned}$$

by (5.1). Hence by (5.3), $\alpha_s r_s < \alpha_s$ and hence $r_s < 1$. But r_s is exactly $\mathbf{E}[|Q'(u)|]$, and the proof is complete. ∎

6 Clutches

In Section 4 we derived differential equations using a heuristic argument for their validity as representing the values of the variables during the course of Process **PColour**(k). In this section, we avoid the assumption about the meaning of the ϕ's and provide estimates of the approximation of the random variables by the differential equation solution.

The prioritisation of types causes a non-smoothness in the process (the expected changes in numbers of variables of different types in one step depends on what type of vertex is chosen as v). We deal with this difficulty using the method introduced in [10] for obtaining bounds on the largest size of an independent set in a random regular graph. For any interval of steps of the process $\mathbf{PColour}(k)$ we are free to define one type (i_0, j_0) to be basic, and decompose the subsequent steps of the process into groups called *clutches* (as in [11]), each clutch consisting of one step in which v is of basic type together with all the steps before the next vertex of basic type is chosen as v. We will only apply the rigorous conclusions of this section when the interval of steps starts at time 0, though being aware that a similar analysis should apply when any given type is first specified as basic should assist the reader in understanding our definitions of the functions ϕ at a later stage.

We wish to consider the process as proceeding clutch by clutch rather than vertex by vertex. This is equivalent to considering the sub-process which takes a shapshot of $\mathbf{PColour}(k)$ at each point in time that a new vertex of basic type is about to be chosen. Thus, we re-define the time variable t to be the number of clutches which have been processed in $\mathbf{PColour}(k)$, and refer to this as the *clutch process* and its steps as *clutch steps* (as opposed to *colouring steps* for $\mathbf{PColour}(k)$). We denote by G_t the (partial) pairing after t clutch steps. As before, (i_0, j_0) is the basic type, B denotes the set of types (i, j) with priority at least $\mathtt{prio}(i_0, j_0)$, and B' denotes those of higher priority than basic. We also set $B^- = \{(i_0, j_0)\}$. This notation will make our equations easier to adapt to modifications in the sequel [7], when there are different types all with the basic priority and chosen amongst with some probability.

For some fixed $\epsilon > 0$, we will consider vectors $\mathbf{Y}$ belonging to a domain $\mathcal{D}_{\epsilon,n}$ defined to be the set of all nonnegative real vectors satisfying

$$S > \epsilon n \tag{6.1}$$

(where S is defined in (4.1)),

$$\sum_{(i,j)\in B'} (d-i) \sum_{|C|=j} \max_{(i',|C'|)\in B'} \beta_{i,C}^{i',C'} < 1 - \epsilon \tag{6.2}$$

where $\beta_{i,C}^{i',C'}$ is the function of $\mathbf{Y}$ defined in (4.3) and also

$$Y_{i,C} > 0 \quad \text{for all } (i, |C|) \in B^-. \tag{6.3}$$

Noting that $\mathbf{Y}$ is determined by a partial pairing G, consider G such that $\mathbf{Y}(G)$ lies in $\mathcal{D}_{\epsilon,n}$. Consider the event that for some t and G, we have $G_t = G$. We consider the very next clutch, $\mathcal{C}$, and all the discussion involving $\mathcal{C}$ is conditional on the event $G_t = G$. The *size* $|\mathcal{C}|$ of $\mathcal{C}$ is the number of steps of $\mathbf{PColour}(k)$ contained in $\mathcal{C}$.

Lemma 6.1 *For some constants $\hat{K}$ and $\hat{c}$ depending only on ϵ,*

$$\mathbf{P}(|\mathcal{C}| \geq s) \leq \hat{K} e^{-\hat{c}s},$$

for all $s < \log^2 n$.

Proof. For the first $\lfloor \log^2 n \rfloor$ colouring steps of the clutch (or for all of them, if the clutch has smaller size), the values of $Y_{i,C}$ can change by at most $O(\log^2 n)$. Similarly, S/n remains bounded away from 0. So $\beta_{i,C}^{i',C'}$ remains the same as its value at the start of the clutch, to within error $o(1)$. Hence we may assume that (6.2) remains true for the first $\lfloor \log^2 n \rfloor$ colouring steps of the clutch.

Each vertex v when it is coloured can create other vertices of the possible types in B' as the pairs are exposed and the neighbouring vertices change type. Vertices which turn into types not in B' we may ignore when investigating the clutch size. For $(i, C) \in B'$, the probability that a vertex of type (i, C) is created due to a pair exposed in an $(i'C')$ vertex is $\beta_{i,C}^{i',C'}$ as given in (4.3) (with error $O(1/S)$). Define $\beta_{i,j}$ to be the maximum in (6.2), divided by $1 - \epsilon$. Then

$$\sum_{(i,j)\in B'} (d - i)\beta_{i,j} < 1. \tag{6.4}$$

For each vertex of type (i', C') coloured, the expected number of new vertices of type (i, C) comes from $d - i'$ Bernoulli trials each with probability at most $\beta_{i,C}^{i',C'}/(1 - \epsilon)$ of success (n sufficiently large). The number of successes is thus stochastically bounded by the binomial random variable $\xi(i', i, C) = \mathrm{Bin}(d - i', \beta_{i,C}^{i',C'}/(1 - \epsilon))$. Summing over $|C| = j$, we conclude that the number of type (i, j) vertices produced is stochastically bounded above by a random variable $\xi(i', i, j)$ with expected value $(d - i')\beta_{i,j}$.

The first colouring step in the clutch colours the first vertex, of basic type, but thereafter we may consider subprocesses, for each of the vertices of type in B' created by that first step. Now compare each of these subprocesses with a multitype branching process P in which the types are (i, j), and the number of children of a type (i', j') individual is distributed as $\xi(i', i, j)$. The number of children is bounded above absolutely and so has an exponentially small tail. We apply Lemma 5.2, letting i in that lemma range over all (i, C) and setting $\alpha_{i',C'} = d - i'$. With (6.4), this implies that the size of P has an exponentially small tail. It follows that the size of the clutch has an exponentially small tail since the clutch size is stochastically bounded above by the size of P. ∎

An immediate corollary of this lemma is that for some $c > 0$

$$\mathbf{P}(|\mathcal{C}| < c \log n) = 1 - o(n^{-1}). \tag{6.5}$$

For each type (i, C), define $\mu_{i,C}$ to be the expected number of vertices of type (i, C) chosen as v in the colouring steps inside the clutch $\mathcal{C}$ (given the partial pairing P_t existing at the start of the t-th clutch). Then by linearity of expectation and by the definition of $r_{i,C}^{i',C'}$, the expected change of $Y_{i,C}$ during a clutch is

$$\mathbf{E}[Y_{i,C}(t + 1) - Y_{i,C}(t) \mid P_t] = \sum_{(i',C')\in B} (d - i')\mu_{i',C'} r_{i,C}^{i',C'} - \mu_{i,C}. \tag{6.6}$$

Now we need to find relations to determine $\mu_{i,C}$. Immediately,

$$\sum_{(i,|C|)\in B^-} \mu_{i,C} = 1 \tag{6.7}$$

as only one vertex of basic type is in each clutch.

Furthermore, for each type (i', C'), let P denote a possible sequence of steps from the beginning of the clutch $\mathcal{C}$, up to part (i) of a colouring step in which a vertex of type (i', C') is chosen as v. (Part (ii) of that step is left unspecified by P.) Call v the *final* vertex of P. Let $\mathcal{P}(i', C')$ be the set of all such sequences P of steps, taken as possible initial subprocesses of the clutch $\mathcal{C}$. For $P \in \mathcal{P}(i', C')$, define $h(P, i, C)$ to be the random variable counting the number of vertices u which become of type (i, C) when the points in the final vertex v of P are paired with one of the points in u. (Note that $h(P, i, C)$ is not determined by P as it refers to the part (ii) of the final colouring step.)

Now let $(i, C) \in B'$. Because of the priority rules, there can be no vertices of type (i, C) at the start of any clutch, and so $Y_{i,C}(t) = 0 = Y_{i,C}(t+1)$. Hence, $\mu_{i,C}$ must exactly equal the number of vertices of type (i, C) which are born during the clutch. This can be expressed as

$$\mu_{i,C} = \sum_{(i',|C'|)\in B} \sum_{P\in\mathcal{P}(i',C')} \mathbf{P}(P)\mathbf{E}[h(P, i, C) \mid P]$$

where the probability refers to a random clutch (still conditional on $G_t = G$ of course). Note that

$$\sum_{P\in\mathcal{P}(i,C)} \mathbf{P}(P) = \mu_{i,C}$$

and, by the definition of $r_{i,C}^{i',C'}$ just above (4.2),

$$\mathbf{E}[h(P, i, C) \mid P] = (d - i')r_{i,C}^{i',C'} + O(\log^2 n/S) \text{ if } P \in \mathcal{P}(i', C'),$$

where the error term $\mathrm{O}(\log^2 n/S)$ is due to the changes in the variables during the clutch, as the sequence P is traced out. Hence we have (ignoring the error term $O(\log^2 n/S)$)

$$\mu_{i,C} = \sum_{(i',|C'|)\in B} \mu_{i',C'}(d - i')r_{i,C}^{i',C'}. \tag{6.8}$$

The solution of these equations for μ will be discussed later. For later convenience, we also examine the number Z of edges exposed in the process. Then $Z = (dn - S)/2$. Since the number of pairs exposed in processing a vertex of type (i, C) is $d - i$, the expected change of Z during a clutch is

$$\mathbf{E}[Z(t+1) - Z(t) \mid P_t] = \sum_{(i,|C|)\in B} (d - i)\mu_{i,C}. \tag{6.9}$$

Equation (6.6) for the expected changes leads us to a system of differential equations for approximating variables, where each $\mathbf{E}[Y_{i,C}(t+1) - Y_{i,C}(t) \mid P_t]$ can be thought of as the derivative $Y'_{i,C}$ (all as functions of the number t of clutches). Now we scale both the number t of clutches and the variables by dividing by n, and approximate $Y_{i,C}/n$ (in a sense to be made precise below) by a real continuous function $y_{i,C}$, and t/n is represented by x. Since the scaling is by the same factor in both cases, each $\mathbf{E}[Y_{i,C}(t+1) - Y_{i,C}(t) \mid P_t]$ approximates the derivative $y'_{i,C}$. Let $\mathbf{y}$ denote the vector

$(y_{i,C})$, $0 \le i < d$, $C \subseteq \{1, \dots, k\}$, and replace Z by z and μ by ν. Then (6.6) and (6.9) become

$$y'_{i,C} = \sum_{(i',|C'|)\in B} (d-i')\nu_{i',C'}(\mathbf{y}) b_{i,C}^{i',C'}(\mathbf{y}) - \nu_{i,C}(\mathbf{y}), \tag{6.10}$$

$$z' = \sum_{(i,|C|)\in B} (d-i)\nu_{i,C}(\mathbf{y}). \tag{6.11}$$

where the functions $b_{i,C}^{i',C'}(\mathbf{y})$ are defined by the equation (4.12), and the equations (6.7) and (6.8) give

$$\sum_{(i,|C|)\in B^-} \nu_{i,C}(\mathbf{y}) = 1, \tag{6.12}$$

$$\nu_{i,C}(\mathbf{y}) = \sum_{(i',|C'|)\in B} (d-i')\nu_{i',C'}(\mathbf{y}) b_{i,C}^{i',C'}(\mathbf{y}) \quad \text{for } (i,|C|) \in B'. \tag{6.13}$$

The initial conditions are

$$y_{0,\emptyset}(0) = 1, \quad y_{i,C}(0) = 0 \text{ otherwise} \tag{6.14}$$

since at the start of the algorithm, $Y_{0,\emptyset}(0) = n$ and $Y_{i,C}(0) = 0$ otherwise.

Note that (6.12) and (6.13) are linear in the ν variables. If we denote the vector of $(\nu_{i,C})$ by ν we may write the equations as

$$M\nu = \mathbf{c}$$

for a suitable coefficent matrix M and a constant vector $\mathbf{c}$. In view of this, and (6.1), (6.2) and (6.3), we will consider for $(x, \mathbf{y})$ the domain defined as

$$\mathcal{D}_{\epsilon,\epsilon_0} = \{(x,\mathbf{y}) : s(\mathbf{y}) > \epsilon,\ F(\mathbf{y}) < 1-\epsilon,\ |y_{i,C}| < 2 \text{ for all } (i,C),\ x > -\epsilon,\ |\det M| > \epsilon_0\}$$

where $F(\mathbf{y})$ is the function in the left side of (6.2) obtained on setting $\mathbf{y}(x) = \mathbf{Y}(nx)/n$. Here $\epsilon > 0$ is fixed as before, and ϵ_0 is any other positive constant.

We say that a type (i,j) is *common* in $\mathbf{PColour}(k)$ (with a given starting point t_0) if, for some $\epsilon' > 0$,

$$\mathbf{P}\left(\sum_{|C|=j} Y_{i,C}(t_0+t) > 0 \text{ for all } 1 \le t \le \epsilon' n\right) = 1 - o(1).$$

For our present concerns, it will be the type $(1,1)$ which is common; in [7] the argument will have to be widened to include other types.

Theorem 6.2 *Let $\tilde{\mathbf{y}}$ be the solution to (6.10) and $\tilde{z}$ the solution to (6.11) with initial conditions given by $y_{0,\emptyset}(0) = 1$, $y_{i,C}(0) = 0$ otherwise, and $z(0) = 0$. Assume that the basic type is common with starting point $t_0 = 0$. Let x_0 be the infimum of all $x > 0$ such that either $(x, \tilde{\mathbf{y}}(x)) \notin \mathcal{D}_{\epsilon,\epsilon_0}$, or $\tilde{y}_{i,C}(x) = 0$ for all basic (i,C). Then there is a function $\lambda = o(1)$ such that*

$$\mathbf{P}[|\mathbf{Y}(t)/n - \tilde{\mathbf{y}}(t/n)| > \lambda \text{ for some } 0 \le t < n(x_0 - \lambda)] = o(1) \tag{6.15}$$

and

$$\mathbf{P}[|Z(t)/n - \tilde{z}(t/n)| > \lambda \text{ for some } 0 \le t < n(x_0 - \lambda)] = o(1).$$

Proof. We start with the proof of (6.15) and then deal with Z. Define

$$\tilde{D} = \big\{(x, \mathbf{y}) \,:\, x = 0 \text{ or } y_{i,C} > 0 \text{ for some } (i, |C|) \in B^-, \;\; y_{i,C} \geq 0 \text{ for all } (i, C)\big\}.$$

This makes the restriction that at least one of the variables corresponding to vertices of basic type is strictly greater than 0 (after the first step). Also, define the stopping time $T_{\tilde{D}}$ to be the minimum $t > 0$ such that $(t/n, \mathbf{Y}(t/n)/n) \notin \tilde{D}$. We apply [11, Theorem 6.1] with this definition of $\tilde{D}$, and with $D = \mathcal{D}_{\epsilon,\epsilon_0}$. The relevance of D to that theorem is via its reference to [11, Theorem 5.1], whose first two hypotheses are required to be verified under the assumptions that $t < T_{\tilde{D}}$. The hypotheses themselves contain the assumption that $t < T_D$. For the first of these, the boundedness hypothesis, we may define $\beta = \log^2 n$ and γ the function which is $o(n^{-1})$ in the right hand side of (6.5). That equation is valid because (6.3) follows from the fact that $t < T_{\tilde{D}}$, and (6.1) and (6.2) follow from $T < T_D$ (where T comes from [11, Theorem 6.1]). The second, the trend hypothesis, we obtain from (6.6). The function λ_1 (in the notation of that theorem) is $o(1)$ and the functions f (actually $f_{i,C}$) are the approximating functions of $\mathbf{E}\Delta Y_{i,C}$ in terms of the scaled functions $y_{i,C}$ given in the right hand side of (6.6). We must also verify the Lipschitz hypothesis, but this comes immediately from the fact that the domain $\mathcal{D}_{\epsilon,\epsilon_0}$ is bounded away from the singularities of the functions $f_{i,C}$, and so they have bounded derivative on $\mathcal{D}_{\epsilon,\epsilon_0}$. One of the singularities of these functions is at $s = 0$ and the only others come from the solutions of the ν's in (6.12) and (6.13). These points do not occur arbitrarily close to $\mathcal{D}_{\epsilon,\epsilon_0}$ because the determinant of M is bounded away from 0 on this domain.

The conclusion of [11, Theorem 6.1] lets us deduce the statement (6.15) with the extra restriction on the range of t, that $t < T_{\tilde{D}}$. We need to remove that condition. The assumption that the basic type is common, together with noting that $y_{i,C} \geq 0$ is always true during the process, lets us remove the condition that $t < T_{\tilde{D}}$, and thus (6.15) follows.

The argument about Z is similar, obtained by including the variable z in the definition of the domain $\mathcal{D}_{\epsilon,\epsilon_0}$ and adjusting the definition of x_0 accordingly. The details of this are entirely straightforward, so omitted. ∎

It is easy to see that in the solution of the differential equations (6.10) with the given initial condition, we have by symmetry, $y_{i,C} = y_{i,D}$, and $\nu_{i,C} = \nu_{i,D}$ whenever $|C| = |D|$. Hence we can simply write $y_{i,j}$ for $y_{i,C}$, and $\nu_{i,j}$ for $\nu_{i,C}$ when $|C| = j$. Then the differential equations (6.10) and (6.11) are simplified to

$$y'_{i,j} = \alpha(\mathbf{y}) b_{i,j}(\mathbf{y}) - \nu_{i,j}(\mathbf{y}), \tag{6.16}$$

$$z' = \alpha(\mathbf{y}) \tag{6.17}$$

where the definitions of $b_{i,j}(\mathbf{y})$ are in (4.14) and

$$\alpha(\mathbf{y}) = \sum_{(i,j)\in B} (d - i) \binom{k}{j} \nu_{i,j}(\mathbf{y}).$$

Noting (6.12) and (6.13), we have that

$$\alpha(\mathbf{y}) = d - i_0 + \alpha(\mathbf{y}) \sum_{(i,j)\in B'} (d - i) \binom{k}{j} b_{i,j}(\mathbf{y}),$$

which implies that

$$\alpha(\mathbf{y}) = (d - i_0)\left(1 - \sum_{(i,j)\in B'} (d-i)\binom{k}{j} b_{i,j}(\mathbf{y})\right)^{-1}. \tag{6.18}$$

This is well defined because of the condition $F(\mathbf{y}) < 1 - \epsilon$ in the definition of $\mathcal{D}_{\epsilon,\epsilon_0}$, and similarly it is clearly positive. Now, noting (6.12) and (6.13), we can write ν in the following way.

$$\nu_{i,j}(\mathbf{y}) = \begin{cases} 1/\binom{k}{j} & \text{if } (i,j) = (i_0, j_0), \\ \alpha(\mathbf{y}) b_{i,j}(\mathbf{y}) & \text{if } (i,j) \in B'. \end{cases} \tag{6.19}$$

This also shows that there are unique solutions on a neighbourhood of the point in question, and since F is Lipschitz, the condition $|\det M| > \epsilon_0$ must hold for sufficiently small $\epsilon_0 > 0$. Thus, for the symmetric solution we may redefine F as

$$F(\mathbf{y}) = \sum_{(i,j)\in B'} (d-i)\binom{k}{j} b_{i,j}(\mathbf{y}), \tag{6.20}$$

and we have the following.

Theorem 6.3 *Let $\tilde{\mathbf{y}}$ be the solution to (6.16–6.19) with initial conditions given by $y_{0,0}(0) = 1$ and $y_{i,j}(0) = 0$ otherwise, and $z(0) = 0$. Define $\tilde{y}_{i,C} = \tilde{y}_{i,j}$ for all C with $|C| = j$. Then the conclusion of Theorem 6.2 holds with x_0 defined as*

$$x_0 = \inf\{x > 0 : s(\tilde{\mathbf{y}}) = \epsilon \text{ or } F(\tilde{\mathbf{y}}) \geq 1 - \epsilon \text{ or } \tilde{y}_{i,j}(x) = 0 \text{ for all basic } (i,j)\} \tag{6.21}$$

where F is defined in (6.20).

For later reference, the condition $F(\mathbf{y}) = 1 - \epsilon$ is loosely referred to as the *explosion* condition: when it holds, the size of a clutch is in danger of having infinite expected size if the process continues for a substantial number of steps (depending on ϵ).

Theorem 6.3 has the following consequence, which we state for later use. Note that the use of a.a.s. $\mathcal{A}$, where $\mathcal{A}$ is a statement involving $o()$, is defined in [13].

Corollary 6.4 *Assume that the basic type is common with starting point 0. Let $\tilde{\mathbf{y}}$ and $\tilde{z}$ be the solution to (6.16–6.19) with initial conditions $y_{0,0}(0) = 1$ and $y_{i,j}(0) = 0$ otherwise, and $z(0) = 0$. Define $\tilde{y}_{i,C} = \tilde{y}_{i,j}$ for all C with $|C| = j$. If $x_0^- < x_0$ as defined in (6.21), then we have that in procedure* **PComplete**$(k, \mathcal{S})$ *with $|\mathcal{S}|$ bounded, a.a.s.*

$$Y_{i,C}(t) = n\tilde{y}_{i,C}(t/n) + o(n) \text{ uniformly for all } i, j, \text{ and all } 0 \leq t < x_0^- n, \tag{6.22}$$

and

$$Z(t) = n\tilde{z}(t/n) + o(n) \text{ uniformly for all } 0 \leq t < x_0^- n. \tag{6.23}$$

Proof. Since the number $|\mathcal{S}|$ of edges in cycles in $\mathcal{S}$ is bounded, the initial conditions for **PColour**(k), as called inside **PComplete**$(k, \mathcal{S})$, satisfy $\overline{Y}_{0,0}(0) = n - O(1)$ and $\overline{Y}_{i,j}(0) = O(1)$ otherwise. So the initial conditions for the corresponding differential

equation (6.16) satisfy $y_{0,0}(0) = 1 - O(1/n)$ and $y_{i,j}(0) = O(1/n)$ otherwise. As $n \to \infty$, these conditions tend to $y_{0,0}(0) = 1$ and $y_{i,j}(0) = 0$, so by a standard result in the theory of ordinary differential equations, the solutions with these differing initial conditions will tend towards each other provided the derivatives are Lipschitz on the domain in question. Note that if $|Y_{i,C} - Y_{i,D}| \leq \epsilon_n$ ($\to 0$ as $n \to \infty$) when $|C| = |D|$, then there is still a unique solution for the ν's when n is sufficiently large. This follows from the fact that they are determined by a linear system with nonzero determinant and changing the entries of the matrix M by a small positive number δ (relying only on ϵ_n), will not make the determinant zero and so the condition $|\det M| > \epsilon$ still holds. For similar reasons, the derivative functions on the right hand side of (6.16) are Lipschitz on the domain we are using. Hence applying Theorem 6.3, the solution for **PComplete**$(k, \mathcal{S})$ is arbitrarily close to the solution for **PColour**(k) on the domain $0 \leq t < x_0^- n$. The assertion of the corollary is established. ∎

In order to get the differential equations (6.16) into the same form as (4.20), we change variables from x to $z = (d - s)/2$. Using (6.17) and the fact noted after (6.18) that $\alpha > 0$,

$$\frac{\mathrm{d}y_{i,j}}{\mathrm{d}z} = \frac{\mathrm{d}x}{\mathrm{d}z} y'_{i,j} = b_{i,j}(\mathbf{y}) - \nu_{i,j}(\mathbf{y})/\alpha(\mathbf{y}).$$

Considering that

$$\nu_{i_0,j_0}(\mathbf{y})/\alpha(\mathbf{y}) = \left(\binom{k}{j_0}\alpha(\mathbf{y})\right)^{-1} = \left(1 - \sum_{(i,j)\in B'} (d-i)\binom{k}{j} b_{i,j}(\mathbf{y})\right)\left((d-i_0)\binom{k}{j_0}\right)^{-1}$$

by (6.18), we obtain (4.20). Note that, compared with the derivation of these equations in Section 4, the derivation in the present section is rigorous. Finally recall that the argument following (4.19) shows that it is equivalent to the system (4.21). It is the latter which we will work with as far as solving the equations go, but the behaviour of clutches will still require some more attention.

7 Type $(1, 1)$ is common

To apply the results of the previous section, we will define priorities so that after the first step of **PColour**(k), the type $(1, 1)$ becomes basic (in the informal sense — see the start of Section 4). We need to show that this type is common, i.e. the number of vertices of type $(1, 1)$ a.a.s. stays positive until a significant number of steps of the process have been taken.

Our object in this section is to prove the following guarantee.

Lemma 7.1 *Asymptotically almost surely, when* **PColour***(k) is applied starting with the empty pairing (or with the pairs in $\mathcal{S}$, as in* **PComplete***$(k, \mathcal{S})$), the type $(1, 1)$ is common.*

We use two lemmas which apply for any fixed starting set $\mathcal{S}$; the proof of the first is immediate. Recall that P is the random pairing in the model of random d-regular graphs discussed in Section 2, and $G(P)$ is the corresponding multigraph.

Lemma 7.2 *If there are no vertices of type (1,1) in* **PColour***(k) at some step, then the set T of coloured vertices at that point is such that every neighbour of T in $G(P)$ is adjacent to at least two vertices of T.*

Supposing that T is such a set, let $a = |T \cup N(T)|$. Note that $|N(T)| \leq d|T|/2$ (by the adjacency condition on the neighbours). This implies that $|T| \geq 2a/(d+2)$, and so $T \cup N(T)$ induces a subgraph of $G(P)$ with at least

$$d|T|/2 + |N(T)| = (d-2)|T|/2 + a \geq 2ad/(d+2) \geq \alpha a$$

edges where $\alpha \geq 4/3$ (as $d \geq 4$). This can be shown not to occur (a.a.s.) using known results such as [12, Lemma 2.14] or Bollobás [3, Theorem 32, Chapter 7], provided $d > 4$. To cover the case $d = 4$ as well we provide the following which is essentially an isoperimetric inequality.

Lemma 7.3 *Let $d \geq 3$ and $\alpha > 1$, and let P be the random pairing in the model of random d-regular graphs. Then for sufficiently small fixed $\epsilon > 0$, it is a.a.s. true that for all $1 \leq a < \epsilon n$, $G(P)$ has no subgraph with a vertices and at least αa edges.*

Proof. If it has such a subgraph then it has an induced one. We compute the expected number of a-sets of vertices inducing such a subgraph. The number of a-sets is $\binom{n}{a}$. For each such set A, the probability that A gives the desired subgraph can be computed as follows. Pair the points in A randomly by picking the next unpaired point from A each time. If this subgraph is to occur, then it is determined after at most $da - \alpha a$ pairs. The probability that at least αa of these random events result in the second point being chosen in A is at most the tail $\mathbf{P}(Z \geq \alpha a)$ where Z is the binomial variable $\mathrm{Bin}(da - \alpha a, a/(n-a))$ since each event has probability at most $a/(n-a)$ of success. We can assume $\alpha \leq d/2$, and hence that ϵ is so small that the tail probability is $O(\mathbf{P}(Z = \lfloor \alpha a \rfloor))$. Thus the expected number of vertex sets in question is at most some constant times

$$\binom{n}{a}\binom{(d-\alpha)a}{\alpha a}\left(\frac{a}{n-a}\right)^{\alpha a}\left(1-\frac{a}{n-a}\right)^{(d-2\alpha)a}.$$

Ignoring the last factor and using the standard bound $\binom{n}{i} \leq (en/i)^i$, this becomes $(O(1)^\alpha (a/n)^{\alpha-1})^a$. For ϵ sufficiently small, this is $o(1)$ when summed over $1 \leq a < \epsilon n$. ∎

Lemma 7.1 follows immediately by combining Lemmas 7.2 and 7.3.

8 Limiting dangerous types

Throughout this section we continue to examine the clutch process, which uses time t to measure the number of clutches processed as in (6.22), rather than the number of vertices coloured.

8.1 Defining t_1

In accordance with the definition in the Introduction, a type (i, C) is called *dangerous* if $|C| \geq k-1$ and $i < d$. We begin with a rough description of the argument required to show that **PComplete**$(k, \mathcal{S})$ will succeed with probability close to 1. For this, we ensure that the probability that a neighbour of a vertex of a dangerous type is coloured at any time is small. To this end, we will need to analyse the algorithm with no dangerous types having basic priority or lower. (Recall that in Section 6 we are free to designate any type as basic.) We also need to condition upon no short cycles being created in **PColour**(k), since a vertex of dangerous type can be joined to another one if a short cycle is "traversed" while one clutch is being processed. Equation (6.5), which tells us that the clutches are a.a.s. small, will then complete the argument.

Recall that the process **PComplete**$(k, \mathcal{S})$ produces successive pairings $P_0, P_1, \ldots$. We will be examining the process for $t \leq t_1$, where $t_1 = \lfloor nx_0^- \rfloor$ with $x_0^- < x_0$ defined in (6.21).

8.2 Path counting

First, we need to know that the numbers of paths of length up to g are well behaved (a.a.s.) throughout the process until time t_1. For any k-coloured path Q (with designated orientation distinguishing the first vertex from the last) of length at most g, and $i, j \leq d$, let $Z_{Q,i,C,i',C'}(t)$ denote the number of copies of Q in the graph $G(P_t) \cup \mathcal{S}$ in which the first vertex has type (i, C) and the last vertex has type (i', C'). At each step in the process, the expected increase in a particular Z variable can be computed to error $O(1/S)$ as a function of these variables. (Note that paths of length 0 are already counted by the $Y_{i,C}$.) Hence, arguing as for the $Y_{i,C}$, we obtain the same kind of sharp concentration result for the $Z_{Q,i,j,i',j'}(t)$ (defined in the obvious way, by taking the symmetric solution) as expressed in (6.22): with probability $1 - o(1/n)$, for $t \leq t_1$

$$Z_{Q,i,j,i',j'}(t) = nz_{Q,i,j,i',j'}(t/n) + o(n), \quad \text{for all } i, j, i', j' \tag{8.1}$$

(where the z variables satisfy a system of differential equations with a bounded solution in the relevant domain). Obtaining this fact is much more straightforward than the analysis of the Y variables, since the Z variables do not influence the choice of the next operation. Note that we do not need to find the solution of these differential equations: we only require knowledge of the sharp concentration of the Z variables.

8.3 Limiting bad pairs in the absence of short cycles

We remind the reader that in all the discussion about graphs of pairings, they are actually pseudographs: loops and multiple edges are permitted. Of course, when conditioning on no short cycles, we may assume $g \geq 3$ and hence that no loops or multiple edges occur. It probably also assists to point out that the event that a short cycle is created using at least one of the edges in $\mathcal{S}$, but not contained in $\mathcal{S}$, has probability $o(1)$ (since it would produce a small subgraph with more edges than vertices so is negligible by Lemma 7.3).

Recall that a bad edge is one which is added during the process **PComplete**$(k, \mathcal{S})$ and joins two vertices which each have $k-1$ different colours on their neighbours (before the edge is added) and that does not create a short cycle. A bad pair is a pair corresponding to a bad edge. The aim of this section is to prove Lemma 8.2 below.

Define the following events for **PColour**(k).

U_t: the event that there are no unwanted cycles in $G(P_t)$,

B_t: the event that there is no bad pair in $G(P_t)$,,

H_t: the event that the approximations in (6.22) for $Y_{i,C}(t)$, and in (8.1) for the $Z_{Q,i,j,i',j'}(t)$, hold in P_t,

L_t: the event that the number of cycles of length less than $2g$ at time t is at most $\log^2 n$,

HL_t: $H_t \cap L_t$.

The event L_t is needed in order to make sure that, when one pair is added, the probability of creating many unwanted cycles is quite small.

We first consider the probability of the event U_t.

Lemma 8.1 *For fixed g, there exists $\xi > 0$ such that $\mathbf{P}(U_t) > \xi + o(1)$ uniformly for all $t > 0$.*

Proof. The asymptotic distribution of cycles of lengths up to g in the standard pairing model is easily found by computing the joint factorial moments of the numbers of cycles of the various lengths (see [12] for example). One conclusion from this is that the probability that there are no cycles of length less than $g+1$ is asympotically equal to a positive constant. In the current setting, we can easily compute these same moments for the numbers of unwanted cycles of lengths $1, \ldots, g$ in the final graph. The only difference is that in the present case, the pairing begins with the pairs in $\mathcal{S}$ as a starting point. This makes no asymptotic difference to the factorial moments, and thus the same conclusion holds. ∎

We now come to the major result of this section.

Lemma 8.2 *Define t_1 as in Section 8.1, and assume that all dangerous types have higher priority than basic. For fixed g and $\mathcal{S}$, in* **PComplete**$(k, \mathcal{S})$ *we have*

$$\mathbf{P}(B_{t_1} \mid U_{t_1}) > 1 - \epsilon_g + o(1)$$

for some constant ϵ_g such that $\epsilon_g \to 0$ as $g \to \infty$.

In proving Lemma 8.2 we may assume (6.22) and also (8.1) from the discussion above. We also have from Lemma 6.1 that the size of a clutch has a $\hat{K}, \hat{c}$ tail for some positive $\hat{K}$ and $\hat{c}$ depending only on x_0^-.

The proof of Lemma 8.2 requires an intermediate result. For some time we will limit the discussion to the event HL_t and compute some probabilities conditional upon this. The following lemma asserts that the probability of creating a short cycle in step $t+1$ is essentially independent of the pairing P_t, up to error $o(1/n)$, and that the probability of a bad edge is small for large g.

Lemma 8.3 *There are bounded functions $p_U(g,t)$ and constants ϵ_g for $g \geq 3$, where $\epsilon_g \to 0$ as $g \to \infty$, such that the following holds. For fixed $g \geq 3$ and any event $Q \subseteq HL_t$ which is determined by P_t (i.e. measurable with respect to the σ-algebra generated by P_t),*

$$\mathbf{P}(Q \cap \overline{B_{t+1}} \cap B_t) \leq (\epsilon_g + o(1))n^{-1}\mathbf{P}(Q \cap B_t) \tag{8.2}$$

for all $t < t_1$, and

$$\mathbf{P}(Q \cap U_{t+1}) = (1 - p_U(g,t)n^{-1} + o(n^{-1}))\mathbf{P}(Q \cap U_t) \tag{8.3}$$

for all $t < t_1$, where the convergence in each $o()$ is uniform over all such t and Q.

Proof. Let P be a pairing in $HL_t \cap U_t$. For most of this proof, we condition on the event $P_t = P$.

Note that the pairs in the clutch added to P_t induce a connected subgraph T of $G(P_{t+1})$. This is because only the first pair in the clutch is added to a vertex of non-dangerous type, and there are no vertices of dangerous type in $G(P_t) \cup \mathcal{S}$.

We consider (8.2) first. Note that a bad pair must form a cycle in T (since a bad pair in P_{t+1} that is not already present in P_t must join vertices of types which can only exist in T, and the part of T already present when the bad pair is added must be connected). This cycle must have length at least $g+1$, by definition of "bad". So T has at least $g+1$ edges, but the number of edges $|E(T)|$ in T has a $\hat{K}, \hat{c}$ tail by the choice of t_1. Conditioning on the pairing P_t being compatible with the event $Q \cap B_t$, the probability that T grows sufficiently large and contains a cycle is at most

$$\sum_{s>g} \hat{K}\exp(-\hat{c}s)\sum_{i\leq s} O(i/n) \leq \hat{K}\sum_{s>g}\exp(-\hat{c}s)O(s^2/n),$$

since the ith edge added to T has probability $O(i/n)$ of forming a cycle in T. Thus (8.2) follows.

For (8.3), we must show that the probability that the next clutch of pairs forms no short cycle, given $P_t = P$, is $1 - p_U(g,t)n^{-1} + o(n^{-1})$ where $p_U(g,t)$ is independent of the particular pairing $P \in HL_t$. The following discussion is conditional upon the event $P_t = P$ for such a pairing P, and takes g as fixed. It is easiest to consider, instead of cycles, nonbacktracking closed walks, by which we mean closed walks in the graph of the pairing that never traverse the same edge in two successive steps in opposite directions. Such a walk is *short* if it has length at most g. We classify these according to the number of different times they enter and leave the edges of T. (T creates no short closed nonbacktracking walks iff it creates no short cycles. These closed walks are easier to deal with because of the following reason. If the edges of a path P_1 are added to the pairing, where P_1 joins the ends of an existing path P_2, we know that a new closed nonbacktracking walk is created but do not know if a cycle is created without examining the internal vertices of P_1 and P_2 for "accidental" intersections.) Let X_0 denote the number of these closed walks contained entirely in the edges of T, X_1 the number entering (and hence leaving) exactly once, and X_2 the number entering at least twice. By Lemma 8.4, proved below,

$$\mathbf{P}(X_2 > 0) = o(n^{-1}) \tag{8.4}$$

(recalling that g is fixed). Hence, we are done if we can show

$$\mathbf{P}(X_0 + X_1 > 0) = p_U(g,t)n^{-1} + o(n^{-1}) \tag{8.5}$$

for a suitable function $p_U(g,t)$, since then upon summing over all pairings P in the event Q, (8.3) follows.

First we show that $\mathbf{E}X_0$ does not depend much on the pairing P (using the approximation guaranteed by $P \in H_t$). To be precise, we show that it is asymptotic to γ_0/n for some constant γ_0 depending only on t and g.

Let J_q denote the number of new short closed nonbacktracking walks contained in T that are formed by adding the qth pair to T (defined to be 0 if T has less than q pairs). Then

$$X_0 = \sum_{q \geq 1} J_q. \tag{8.6}$$

While T is being generated in the algorithm, the mate for any designated point in a selected vertex v in the pairing algorithm is randomly chosen from all remaining unpaired points. Recall that there are at least $\Omega(n)$ unpaired points in P_t (since $t < t_1 \leq x_0^- n$, and by the definition of x_0^-). Also, there are at most d^g walks of length at most $g-1$ in the current graph starting from v. The end vertex of each of these has at most d unpaired points. Hence, if $q = o(n)$, then $\Omega(n)$ unpaired points still remain, and

$$\mathbf{E}J_q = O(n^{-1})\mathbf{P}(|E(T)| \geq q) \tag{8.7}$$

where the constant implicit in $O()$ is uniform over q (recalling that d and g are fixed). On the other hand, to take care of the case $q \neq o(n)$, for $q > \log^2 n$, we may use

$$\mathbf{E}J_q = O(1)\mathbf{P}(|E(T)| \geq q) = o(n^{-2}) \tag{8.8}$$

since $|E(T)|$ has an exponentially small tail.

We first consider the contribution to (8.6) for moderately large q. For any fixed $R > 0$, (8.7) and (8.8) imply that

$$\begin{aligned} \sum_{q>R} \mathbf{E}J_q &= \sum_{q > \log^2 n} o(n^{-2}) + O(n^{-1}) \sum_{R < q \leq \log^2 n} q\mathbf{P}(|E(T)| = q) \\ &= O(n^{-1})\epsilon_R \end{aligned} \tag{8.9}$$

for some constant ϵ_R, where

$$\epsilon_R \to 0 \text{ as } R \to \infty \tag{8.10}$$

since $|E(T)|$ has an exponentially small tail. (Do not let this observation fool you: R is still fixed and we don't want it moving yet!) Also, the constant implicit in $O()$ is uniform over R (and depends only on $\hat{K}$ and $\hat{c}$, therefore only on t_1).

We comment here that another version of the argument leading to (8.9) is used in the proof of Lemma 8.4. Note that J_q includes closed walks which use edges not in T, so (8.9) will also be useful in analysing X_1.

The cases $q \leq R$ are analysed differently. Consider a fixed connected graph T^* of at most R edges in which each vertex v is labelled with a type $\tau(v) = (i,j)$ for some i and

j, and the edges uv of T^* are directed and labelled by some bijection σ to the numbers $1, 2, \ldots, |E(T^*)|$. Write $J(T^*)$ for the number of closed nonbacktracking walks of length at most g contained in T^*. Then

$$\sum_{q \le R} \mathbf{E}J_q = \sum_{T^*} J(T^*)\mathbf{P}(T_R \equiv T^*) \tag{8.11}$$

where T_R is the graph induced by the first R pairs added to T (or all of T in the case it has less than R pairs), and $\equiv$ denotes that there is an isomorphism θ from T^* to T_R taking each vertex v to one whose type in P_t is $\tau(v)$, such that the labelling σ of pairs in T^* is the order that the corresponding pairs in T_R are added by the algorithm, and such that the edge is directed away from the vertex chosen as v in the step in **PColour**(k) in which that pair is added.

Now $\mathbf{P}(T_R \equiv T^*)$ can be evaluated as follows. Let r denote the number of vertices of T^*. Using the approximation expressed by H_t, there is a bounded function $\alpha(t, T^*)$ (recalling that T^* defines τ) such that the number of ways to choose vertices of P_t of appropriate types for T^* is $(\alpha+o(1))n^r$. Then, given such a set of vertices, the probability that they are indeed the vertices of T_R, and that the pairs of T_R are added by the algorithm in the order specified by σ and in the manner specified by their orientations, is $(\beta + o(1))n^{-|E(T^*)|-1}$ for some bounded function $\beta(t, T^*)$. The determination of this formula is a little subtle but straightforward. The -1 occurs in the exponent because the first edge of T_R appears in the place it does with probability $(c + o(1))/n^2$ for some bounded c: first a vertex of the appropriate type is selected (with probability $(c+o(1))/n$, where c depends on its type, as specified by τ), and then for this to be paired with a given vertex introduces a similar factor. On the other hand, for all subsequent edges of T_R, the first vertex of the pair is already determined (or selected uniformly at random from all vertices of T currently of the appropriate type), and it is only the choice of a mate which introduces the extra $1/n$ factor. The reason that β depends on τ is basically that the type of a given vertex determines the probability (asymptotically) that a point in that vertex is the one chosen when the second point of a pair is selected at random from all unpaired points.

It now follows that

$$\mathbf{P}(T_R \equiv T^*) = \big(\alpha(t, T^*)\beta(t, T^*) + o(1)\big)n^{r-|E(T^*)|-1}. \tag{8.12}$$

For $J(T^*) \ge 1$ we have $r \le |E(T^*)|$, and so (8.11) gives

$$\sum_{q \le R} \mathbf{E}J_q = \big(\gamma_0(R, t) + o(1)\big)n^{-1}.$$

where $\gamma_0(R, t)$ is the sum of all products $\alpha\beta$ for graphs T^* with $r \le |E(T^*)| \le R$.

Combining with (8.9) and (8.6) and using linearity of expectation,

$$\mathbf{E}X_0 = \big(\gamma_0(R, t) + O(\epsilon_R)\big)n^{-1}.$$

where the constant implicit in $O()$ depends only on t_1. This statement, being true for all R, implies by (8.10) that $\gamma_0(R, t)$ has a limit $\gamma_0(t)$ as $R \to \infty$ (only here does R start moving) and that

$$\mathbf{E}X_0 = \big(\gamma_0(t) + o(1)\big)n^{-1}. \tag{8.13}$$

Moreover, there is an upper bound on $\gamma_0(R,t)$ that depends only on t_1, and it follows that $\gamma_0(t)$ is bounded by a function of t_1.

The next aim is to show that $\mathbf{E}X_1$ has roughly the same value γ_1/n for all P under consideration. Note that

$$\mathbf{E}X_1 = \sum_{u,v\in V(G)} \sum_{i<g} a_{u,v}(i)p_{u,v}(i) \tag{8.14}$$

where $a_{u,v}$ is the number of nonbacktracking (u,v)-walks of length i in $G(P_t) \cup \mathcal{S}$ and $p_{u,v}$ denotes the expected number of nonbacktracking walks in T between u and v of length at most $g-i$. For such a walk to exist, u and v must be vertices of T, which for any particular u and v will turn out to have probability $O(n^{-1})$.

We now use a related approach as for X_0, fixing first an R. Since J_q includes closed walks which use edges not in T as well as those in T, (8.9) can be used as before for the short closed nonbacktracking walks created after the first R edges of T. This applies to all paths between u and v in T which use any edge outside of the graph T_R defined above. So consider the contribution to $p_{u,v}(i)$ from nonbacktracking (u,v)-walks contained in T_R. This can be written as

$$\sum_{T^*} \sum_{u^*,v^*\in V(T^*)} \bar{a}_{u^*,v^*}(i)p(u,u^*,v,v^*,T^*)$$

where $\bar{a}$ denotes the number of nonbacktracking (u^*,v^*)-walks in T^* of length at most $g-i$, and $p(u,u^*,v,v^*,T^*)$ is the probability that $T_R \equiv T^*$ (defined as for X_0) with $\theta(u^*)=u$ and $\theta(v^*)=v$. By the sharp concentration of the variables Z implied by H_t, and arguing as for (8.12),

$$p(u,u^*,v,v^*,T^*) = \big(\alpha'(t,u^*,v^*,T^*) + o(1)\big)n^{r-|E(T^*)|-3}.$$

(The extra factor n^{-2} comes from the probability that, of all vertices of the same types as u and v, these are the particular vertices selected, is of the form $\big(c+o(1)\big)/n^2$.) Thus, the cofactor of $a_{u,v}(i)$ in (8.14) depends essentially only on the types of u and v. On the other hand, the sum of $a_{u,v}(i)$ over all u and v of given types is, by (8.1), equal to $\big(\beta'(t)+o(1)\big)n$ for some function β'. Thus

$$\mathbf{E}X_1 = \big(\gamma_1(t)+o(1)\big)n^{-1} \tag{8.15}$$

for a function $\gamma_1(t)$ which one can easily check is bounded by a function of t_1.

Next, we observe that the expected number of *ordered pairs* of distinct short nonbacktracking closed walks created by T is $o(n^{-1})$ by Lemma 8.4 below. Hence, putting $X = X_0 + X_1$, we have $\mathbf{E}[X(X-1)] = o(n^{-1})$ and so $\mathbf{P}(X>0) = \mathbf{E}X + o(n^{-1})$. Thus (8.13) and (8.15) imply that

$$\mathbf{P}(X_0+X_1>0) = \big(\gamma_0(t)+\gamma_1(t)+o(1)\big)n^{-1}$$

which gives (8.5) as required, with $p_U = \gamma_0(t)+\gamma_1(t)$. ∎

In that proof we used the following.

Lemma 8.4 *Suppose $P \in HL_t$, and g is fixed. Then the expected number of ordered pairs of distinct short nonbacktracking closed walks created by T is $o(n^{-1})$.*

Proof. Using the approach leading to (8.9), the number of ordered pairs in question is

$$\sum_{q_1 \neq q_2} J_{q_1} J_{q_2} \; + \; \sum_q J_q(J_q - 1). \tag{8.16}$$

As with the derivation of (8.7) we find that

$$\mathbf{E} J_{q_1} J_{q_2} = O(n^{-2}) \mathbf{P}(|E(T)| \geq \max\{q_1, q_2\}).$$

(Note that J_{q_1} and J_{q_2} are not independent, but that does not affect the argument: when adding a pair to T, the mate of a chosen point is selected u.a.r. from all remaining unpaired points, conditional upon the pairs already added to T.)

Putting $q = \max\{q_1, q_2\}$ and rearranging gives

$$\sum_{q \geq 1} O(n^{-2}) q \mathbf{P}(|E(T)| \geq q) = O(n^{-2}) \sum_{q \geq 1} q^2 \mathbf{P}(|E(T)| = q) = O(n^{-2})$$

(c.f. (8.9)). This takes care of the first summation in (8.16).

The summand in the second summation is 0 unless $J_q \geq 2$ and is bounded (by a function of d and g) in any case, so it is enough to show that

$$\sum_q \mathbf{P}(J_q \geq 2) = o(n^{-1}). \tag{8.17}$$

Since $|E(T)|$ has an exponentially small tail, $\mathbf{P}(J_q \geq 2) = o(n^{-2})$ for $q > \log^2 n$ and

$$\sum_{q > \log^2 n} \mathbf{P}(J_q \geq 2) = o(n^{-1}). \tag{8.18}$$

But $J_q \geq 2$ can only happen if the qth edge of T joins two vertices in a nonbacktracking closed walk of length at most $2g - 2$. Since $P \in L_t$, there are $O(\log^2 n)$ such pairs of vertices in the graph $G(P)$ (recalling that g is fixed). After adding the first $q - 1$ pairs of T, the bound $O(\log^2 n)$ may increase to at most $O(q + \log^2 n)$. ∎

Proof of Lemma 8.2 Recalling that the function p_U in (8.3) depends only on g and t, we compute (with all probabilities conditional upon HL_t, which is suppressed for clarity)

$$\begin{aligned}
\mathbf{P}(B_t \mid U_t) &= \frac{\mathbf{P}(B_t \cap U_t)}{\mathbf{P}(U_t)} \\
&= (1 + o(n^{-1})) \frac{\mathbf{P}(B_t \cap U_{t+1})}{\mathbf{P}(U_{t+1})} \qquad \text{applying (8.3) twice} \\
&= (1 + o(n^{-1})) \frac{\mathbf{P}(B_{t+1} \cap U_{t+1}) + \mathbf{P}(\overline{B_{t+1}} \cap B_t \cap U_{t+1})}{\mathbf{P}(U_{t+1})} \\
&= \mathbf{P}(B_{t+1} \mid U_{t+1}) + O(\epsilon_g / n)
\end{aligned}$$

since

$$\begin{aligned}\frac{\mathbf{P}(\overline{B_{t+1}} \cap B_t \cap U_{t+1})}{\mathbf{P}(U_{t+1})} &\leq \frac{\mathbf{P}(\overline{B_{t+1}} \cap B_t \cap U_t)}{\mathbf{P}(B_t \cap U_{t+1})} \\ &\leq (1+o(1))\frac{\mathbf{P}(\overline{B_{t+1}} \cap B_t \cap U_t)}{\mathbf{P}(B_t \cap U_t)} \qquad \text{by (8.3)} \\ &= O(\epsilon_g/n)\end{aligned}$$

by (8.2). Thus

$$\mathbf{P}(B_{t+1} \mid U_{t+1} \cap H_t) = \mathbf{P}(B_t \mid U_t \cap H_t) - O(\epsilon_g/n). \tag{8.19}$$

On the other hand, it is easy to see that this implies the same result without conditioning on HL_t, using the facts (shown below) that $\mathbf{P}(HL_t) = 1-o(n^{-1})$ and that $\mathbf{P}(U_t)$ is bounded below by a positive constant. For then $\mathbf{P}(Q \cap HL_t) = \mathbf{P}(Q) + o(n^{-1})$ for all events Q, such as $B_{t+1} \cap U_{t+1}$. With this new unconditional interpretation of (8.19), we obtain by summing over all $t < t_1$

$$\mathbf{P}(B_{t_1} \mid U_{t_1}) = \mathbf{P}(B_0 \mid U_0) - O(\epsilon_g) = 1 - O(\epsilon_g) \tag{8.20}$$

as required. The properties of ϵ_g required in the lemma follow from its definition in Lemma 8.3.

It remains to show the two facts used above. Of course, $\mathbf{P}(H_t) = 1 - o(n^{-1})$ was shown at (6.22). One can show $\mathbf{P}(L_t) = 1 - o(n^{-1})$ by calculating the $\lceil \log^2 n \rceil$ factorial moment of the number of cycles of length at most $2g$ in a random d-regular graph, and finding that it is $o(1/n)$. (See [12] for similar calculations). Finally, we know from Lemma 8.1 that $\mathbf{P}(U_t)$ is bounded below by a positive constant which depends only on d. ∎

9 Succeeding after t_1.

In this section, the aim is to show the following two lemmas concerning the process after time t_1 (with t still referring to the number of clutches). The first asserts that, loosely speaking, the process succeeds with probability close to 1 provided there are no unwanted cycles, and the second bounds the probability that unwanted cycles occur. We extend the definition of U_t to include U_∞, the event that there are no unwanted cycles in the final graph of the process.

Lemma 9.1 *Suppose that $k \geq 3$, and all dangerous types have higher priority than basic, and choose a fixed $\hat{\epsilon} > 0$. Consider the graph G_{t_1} at a time t_1 in* **PComplete***$(k, \mathcal{S})$. Assume that the process has not failed by time t_1, and that G_{t_1} has no unwanted cycle and has ν_i vertices of degree $d-i$ for each i. Assume furthermore that*

$$\frac{\sum_i i(i-1)\nu_i}{\sum_i i\nu_i} < 1 - \hat{\epsilon}. \tag{9.1}$$

Then, conditional upon the event U_∞, the process succeeds with probability at least $1 - \epsilon_g + o(1)$, where ϵ_g is constant for fixed g and $\hat{\epsilon}$, and moreover $\epsilon_g \to 0$ as $g \to \infty$.

Proof. The graph G^* induced by the pairs added after the formation of G_{t_1} at time t_1 has ν_i vertices of degree i. The expected number of cycles of length r in G^* is given asymptotically by [9, Theorem 4(ii) and the second equation after (4.10)] as $(2\alpha)^r/(2r)$, in which 2α is the expression on the left side of (9.1). So it is asymptotically at most $(1-\hat{\epsilon})^r/2r$ (as $n \to \ infty$). This also remains true if one conditions on no unwanted cycles occurring at all after t_1. To see this, consider computing the joint factorial moments of the numbers of unwanted cycles of lengths $1, \ldots, g$ and r. The calculations are straightforward, a modification of what is required in the proof of Lemma 8.1, with the extra twist that the moments depend (in a simple way) on the numbers of paths in G_{t_1} of given degree endvertices. The calculation shows that these variables are asymptotically independent Poisson, from which the claim follows.

Thus, conditioning on G^* having no unwanted cycles, the expected number of cycles in G^* is at most $o(1) + \sum_{r>g}(1-\hat{\epsilon})^r/2r$. As long as ϵ_g is larger than this sum, we have that G^* is acyclic with probability at least $1 - \epsilon_g + o(1)$. When G^* is acyclic, it can always be 2-list-coloured, and hence the required colouring of G^* exists. Moreover, the fact that dangerous types have higher priority than all other types ensures that the process succeeds in all cases that an acyclic G^* is produced. ∎

The second lemma tells us that the probability of unwanted cycless is more or less independent of whether the process has failed by time t_1.

Lemma 9.2 *Suppose $k \geq 3$. There is a function p, bounded away from 0, such that the following holds. Consider a graph G such that $H_{t_1} \cap U_{t_1}$ is satisfied if $G_{t_1} = G$. Then $\mathbf{P}(U_\infty \mid G_{t_1} = G) = p(t_1/n) + o(1)$, uniformly over such G.*

Proof. Let $X_i(G)$ denote the number of unwanted cycles of length i created in the process by the edges added after time t_1, given $G_{t_1} = G$. It is routine to compute the joint factorial moments of $X_1(G), \ldots, X_g(G)$ as a function of the numbers of short paths of G (a calculation similar to that referred to in the proof of the previous lemma). We conclude using the usual moment method for Poisson approximation (again, see [12] for a similar argument) that $\mathbf{P}(U_\infty \mid G_{t_1} = G) = e^{-\mathbf{E}X} + o(1)$, where $X = \sum_{i=1}^{g} X_i$. It is also routine to check, using the sharp concentration implied by H_{t_1}, that $\mathbf{E}X = p + o(1)$ for a constant p depending only on t_1/n. The asymptotics in $o(1)$ in such a situation is uniform over all graphs G provided their maximum degree is bounded. The lemma follows. ∎

10 Wrapping it up

In this section we put the results in the previous three sections together to distil the main result to be used in conjunction with the differential equation solution. The result is stated in a form convenient for use with $d = 4$ but easily modified for arbitrary d. The use of $x_0^- < x_0$ is just a convenient way to make it clear that the differential equations do not need to be solved as far as x_0; finding such a value x_0^- will suffice. In this section, any mention of time refers to measurement in terms of the real scaled time for the clutch differential equations.

Theorem 10.1 *Suppose that in* **PComplete**$(k,\mathcal{S})$*, all dangerous types have higher priorities than all other types. Let the type* $(1,1)$ *be the only basic type for the purpose of defining* $y_{i,j}$ *as the solution of (6.16) with initial conditions as in (6.3), and* x_0 *in (6.21). Defining* x_0 *as in (6.21), suppose that*

$$\frac{\sum_{i,j}(d-i)(d-i-1)\binom{k}{j}y_{i,j}(x_0^-)}{\sum_{i,j}(d-i)\binom{k}{j}y_{i,j}(x_0^-)} < 1 \tag{10.1}$$

for some $x_0^- < x_0$*. Then for some function* $g = g(n)$ *which goes to infinity sufficiently slowly as* $n \to \infty$*, the algorithm* **Short**(g,k) *a.a.s. succeeds when applied to a random* $G \in \mathcal{G}_{n,d}$*. Consequently, such a graph is a.a.s.* k*-colourable.*

Proof. Let $t = \infty$ to denote the final time. Consider the process **PComplete**$(k,\mathcal{S})$ for fixed g, k and $\mathcal{S}$. We first wish to bound the probability that the algorithm fails, conditional upon U_∞ (a condition which we can remove later). We set $t_1 = \lfloor nx_0^- \rfloor$. By Lemma 8.2, $\mathbf{P}(\overline{B_{t_1}} \mid U_{t_1}) < \epsilon_g + o(1)$ where $\epsilon_g \to 0$ as $g \to \infty$.

Combining this with Lemma 8.1, we have that for sufficiently small $\xi > 0$

$$\begin{aligned}\mathbf{P}(U_{t_1} \cap B_{t_1}) &= \mathbf{P}(B_{t_1} \mid U_{t_1})\mathbf{P}(U_{t_1}) = (1 - \mathbf{P}(\overline{B_{t_1}} \mid U_{t_1}))\mathbf{P}(U_{t_1}) \\ &> \big(1 - \epsilon_g - o(1)\big)\big(\xi + o(1)\big) \geq \xi - O(\epsilon_g).\end{aligned}$$

This shows that $\mathbf{P}(U_{t_1} \cap B_{t_1})$ is asymptotically at least some fixed positive constant, for sufficiently large g. Conditional upon $U_{t_1} \cap B_{t_1}$, the process does not fail by time t_1. It thus follows from (6.22) in Corollary 6.4 and Lemma 7.1 that G_{t_1} also a.a.s. satisfies the condition (9.1) for some $\hat{\epsilon} > 0$. Hence, by Lemma 9.1, conditional upon U_∞ as well, **PComplete**$(k,\mathcal{S})$ succeeds with probability at least $1 - \epsilon_g + o(1)$ (for another such function ϵ_g). Writing B for the event that the process succeeds, we thus have $\mathbf{P}(\overline{B} \mid U_\infty \cap B_{t_1}) < \epsilon_g + o(1)$ and thus

$$\mathbf{P}(\overline{B} \cap B_{t_1} \mid U_\infty) = \frac{\mathbf{P}(\overline{B} \cap B_{t_1} \cap U_\infty)}{\mathbf{P}(U_\infty)} \leq \frac{\mathbf{P}(\overline{B} \cap B_{t_1} \cap U_\infty)}{\mathbf{P}(U_\infty \cap B_{t_1})} = \mathbf{P}(\overline{B} \mid U_\infty \cap B_{t_1}) < \epsilon_g + o(1). \tag{10.2}$$

We also know from Lemma 9.2 that the number of unwanted cycles created after time t_1 is more or less independent of B_t (or on any particular graph G_{t_1}). Specifically,

$$\mathbf{P}(U_\infty \mid A \cap H_{t_1} \cap U_{t_1}) = p(t_1/n) + o(1).$$

for $A = U_{t_1}$ and also for $A = \overline{B_{t_1}}$. Noting by (6.22) that $\mathbf{P}(H_{t_1}) = 1 - o(1)$, this implies $\mathbf{P}(U_\infty \cap A) = p(t_1/n)\mathbf{P}(U_{t_1} \cap A) + o(1)$ since p is bounded below and U_{t_1} contains U_∞. Thus

$$\mathbf{P}(U_\infty) = p(t_1/n)\mathbf{P}(U_{t_1}) + o(1), \tag{10.3}$$

$$\mathbf{P}(U_\infty \cap \overline{B_{t_1}}) = p(t_1/n)\mathbf{P}(U_{t_1} \cap \overline{B_{t_1}}) + o(1). \tag{10.4}$$

We also have $\mathbf{P}(\overline{B_{t_1}} \mid U_{t_1}) < \epsilon_g + o(1)$ from Lemma 8.2, which together with (10.4) gives

$$\mathbf{P}(U_\infty \cap \overline{B_{t_1}}) < p(t_1/n)\epsilon_g\mathbf{P}(U_{t_1}) < \epsilon_g\mathbf{P}(U_\infty) + o(1)$$

by (10.3). As noted in the proof of Lemma 8.2, $\mathbf{P}(U_\infty)$ is bounded below by a positive constant (depending only on d). So this last inequality gives

$$\mathbf{P}(\overline{B_{t_1}} \mid U_\infty) < \epsilon_g + o(1).$$

Combining this with (10.2) produces

$$\mathbf{P}(\overline{B} \mid U_\infty) < 2\epsilon_g + o(1). \tag{10.5}$$

Finally, we need to remove the conditioning on the set $\mathcal{S}$ and on the event U_∞. Here k and d are fixed, and g is fixed in this argument until the very end. Let $\mathcal{S}(G)$ denote the set of short cycles of $G \in \mathcal{G}_{n,d}$, let κ denote the event that $\mathbf{Short}(g,k)$ succeeds, and let $\kappa(\mathcal{S})$ denote the event that $\mathbf{PComplete}(k,\mathcal{S})$ succeeds. Then

$$\mathbf{P}(\kappa) = \sum_{\mathcal{S}} \mathbf{P}(G \in \kappa(\mathcal{S}) \mid \mathcal{S}(G) = \mathcal{S})\, \mathbf{P}(\mathcal{S}(G) = \mathcal{S}). \tag{10.6}$$

By the known distribution of short cycles in $G \in \mathcal{G}_{n,d}$, $\mathbf{P}(|\mathcal{S}(G)| > s) < \epsilon_s + o(1)$ where $\epsilon_s \to 0$ as $s \to \infty$ and this denotes a different sequence of constants from ϵ_g. Choosing s so large that $\epsilon_s < \epsilon_g$, the terms in the summation above with $|\mathcal{S}| > s$ sum to at most $\epsilon_g + o(1)$. For each of the other terms, the conditional probability $\mathbf{P}(G \in \kappa(\mathcal{S}) \mid \mathcal{S}(G) = \mathcal{S})$ is just $\mathbf{P}(B \mid U_\infty)$ for $\mathbf{PComplete}(k,\mathcal{S})$ as bounded in (10.5). Thus (10.6) implies

$$\mathbf{P}(\kappa) > 1 - 3\epsilon_g + o(1).$$

Up until this point, g has been fixed in all our arguments. It is finally time to consider it as a function of n. Since the last inequality is true for all fixed g, but ϵ_g can be made arbitrarily small by taking large g, the theorem follows. (This last step uses the Fact in Section 1, applied with $f(n,g) = \max\{1 - 3\epsilon_g - \mathbf{P}(\kappa), 0\}$.) ∎

11 Solution for $d = 4$, $k = 3$

In this section we prove Theorem 1.1 using Theorem 10.1. For $d = 4$ and $k = 3$, we assign the priorities as follows.

$$\texttt{prio}(0,0) = 0,\ \texttt{prio}(1,1) = 2,\ \texttt{prio}(2,1) = 1,\ \texttt{prio}(3,1) = -1,$$
$$\texttt{prio}(2,2) = 4,\ \texttt{prio}(3,2) = 3.$$

The reason for using these priorities should be self-evident, except for the priority of $(3,1)$, which is set below 0 since such vertices, if they appear, help to reduce the value of the left hand side of the inequality (10.1) in Theorem 10.1. Thus, it is desirable not to colour them. Under this list of priorities, in the informal discussion at the start of Section 4, the type (1,1) is basic at the start of the process $\mathbf{PComplete}(k,\mathcal{S})$. Note that in Theorem 10.1, $(1,1)$ is regarded as basic in order to define the function $\mathbf{y}$.

Recall, as observed after the proof of Corollary 6.4, that the differential equation eqnsde is equivalent to (4.21), which is $\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}v} = A\mathbf{u} + \mathbf{c}$. By (4.22), the coefficient

matrix is

$$A = \begin{pmatrix} -2 & 0 & 0 & 0 \\ 4/3 & 1/3 & 4/9 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2/3 & 1 \end{pmatrix},$$

and the constant vector is $\mathbf{c} = (0, -1/9, 0, 0)$. The standard solution method uses the eigenvalues of the coefficient matrix A. In this case, these are -2, 1, a and b, where $a = (1 + \sqrt{17})/6$ and $b = (1 - \sqrt{17})/6$. The initial condition is

$$\mathbf{u}(0) = \mathbf{y}(0)/s(\mathbf{y}(0)) = \mathbf{y}(0)/d = (1/4, 0, 0, 0).$$

The solution of the differential equation is

$$\begin{aligned}
u_{0,0}(v) &= e^{-2v}/4, \\
u_{1,1}(v) &= \left(\frac{3}{38} - \frac{41\sqrt{17}}{1938}\right) e^{av} + \left(\frac{3}{38} + \frac{41\sqrt{17}}{1938}\right) e^{bv} - \frac{3}{19} e^{-2v}, \\
u_{2,1}(v) &= \left(\frac{97\sqrt{17}}{2584} - \frac{25}{152}\right) e^{av} - \left(\frac{97\sqrt{17}}{2584} + \frac{25}{152}\right) e^{bv} + \frac{3}{38} e^{-2v} + \frac{1}{4}, \\
u_{3,1}(v) &= \frac{1}{3876}\left((357 - 45\sqrt{17}) e^{av} + (357 + 45\sqrt{17}) e^{bv} - 68 e^{-2v} - 646\right).
\end{aligned}$$

Letting v_0 be the first solution of $u_{1,1}(v) = 0$, we solve to obtain

$$\begin{aligned}
v_0 &= 2.1530, \\
u_{0,0} &= 0.0034, \\
u_{1,1} &= 0, \\
u_{2,1} &= 0.0860, \\
u_{3,1} &= 0.1568.
\end{aligned}$$

It is straightforward to verify that, for this solution, $F(\mathbf{y})$ in condition (6.21) stays below 1 for $v \le v_0$, and hence x_0 corresponds to v_0 for ϵ sufficiently small. If the algorithm is permitted to run up to this point, the interpretation is that the vertices of type (1,1) will disappear and some of the next priority down, type $(2, 1)$, will be chosen as v. However, noting that

$$\begin{aligned}
\frac{\sum_{i,j}(d-i)(d-i-1)\binom{k}{j} y_{i,j}}{\sum_{i,j}(d-i)\binom{k}{j} y_{i,j}} &= \sum_{i,j}(d-i)(d-i-1)\binom{k}{j} y_{i,j}/s \\
&= \sum_{i,j}(d-i)(d-i-1)\binom{k}{j} u_{i,j} \\
&= 4 \times 3 \times 0.0034 + 2 \times 3 \times 0.0860 \\
&= 0.5568 < 1
\end{aligned}$$

and that both dangerous types (2,2) and (3,2) have higher priorities than all other types, Theorem 10.1 implies that 4-regular graphs are a.a.s. 3-colourable. This proves

Theorem 1.1, and moreover, Theorem 10.1 provides the simple algorithm which a.a.s. successfully colours a random 4-regular graph with three colours.

We make the final observation that for $d \geq 5$ this argument does not suffice for the particular values of k which are of interest. We are forced to use $(1, 1)$ as the basic type, and regardless of the priorities, the explosion condition (defined after Theorem 6.3) occurs before the density condition (10.1) holds. We address this problem in the sequel [7].

References

[1] D. Achlioptas and C. Moore, Almost all graphs with average degree 4 are 3-colourable, *Journal of Computer and System Sciences*, **67 (4)** (2003), 441-471.

[2] D. Achlioptas and A. Naor, The two possible values of the chromatic number of a random graph, *STOC*, 2004.

[3] B. Bollobás, **Random graphs**. Academic Press, New York, 1985.

[4] J. Díaz, M. Serna and N.C. Wormald, Computation of the bisection width for random d-regular graphs, *Theoretical Computer Science* (to appear).

[5] S. Janson, T. Łuczak and A. Ruciński, **Random graphs**, Wiley, New York, 2000.

[6] M. Molloy and B. Reed, The size of the giant component of a random graph with a given degree sequence, *Combinatorics, Probability and Computing* **7** (1998), 295–305.

[7] L. Shi and N. Wormald, Colouring random d-regular graphs (in preparation).

[8] J. Spencer and N. Wormald, Birth control for giants, manuscript.

[9] N.C. Wormald, The asymptotic distribution of short cycles in random regular graphs, *J. Combinatorial Theory (Series B)*, **31** (1981), 168–182.

[10] N.C. Wormald, Differential equations for random processes and random graphs, *Ann. Appl. Probab.* **5** (1995), 1217–1235.

[11] N.C. Wormald. The differential equation method for random graph processes and greedy algorithms. In M. Karoński and H. Prömel, editors, *Lectures on Approximation and Randomized Algorithms*, pp. 73-155. PWN, Warsaw, 1999.

[12] N.C. Wormald, Models of random regular graphs, In *Surveys in Combinatorics, 1999*, J.D. Lamb and D.A. Preece, eds, pp. 239–298.

[13] N.C. Wormald, Random graphs and asymptotics. Section 8.2 in *Handbook of Graph Theory*, J.L. Gross and J. Yellen (eds), pp. 817–836. CRC, Boca Raton, 2004.