The cook-book approach to the differential equation method*

J. Díaz[1] D. Mitsche[2†]

[1] Llenguatges i Sistemes Informàtics, UPC

[2] Centre de Recerca Matemàtica, UAB

diaz@lsi.upc.edu, dmitsche@crm.cat

Abstract

We survey several applications of the differential equation method in different areas of discrete mathematics. We give examples of its use in the analysis of algorithms in random graph processes and random boolean formulae. We also briefly review the basic theorem of Wormald [77] used in the analysis, but we aim for simplicity and not for maximal generality. The primary goal of this survey is to be a toolbox for the usage of the differential equation method.

1 Introduction and Motivation

Probabilistic techniques play an important role in the fields of algorithmics, complexity and discrete mathematics. For example, if a random variable is defined to measure the running time either of a randomized algorithm or of a deterministic algorithm where the input is taken randomly from a given distribution, we might want to compute not only the expectation of the random variable, but also *concentration*; that is, how much the random variable can deviate from the expectation. Recently, there has been published a nice book devoted to the topic of concentration for computer scientists [30]. Moreover, there is a large collection of books on probability for discrete mathematics and computer science, which deal extensively with techniques for estimating concentration bounds [7, 66, 67, 68, 49]. Also, there have been several general surveys on concentration bounds specifically addressed to researchers in the mentioned fields, among others [58, 59, 45, 28, 29, 16]. However, none of the previous books and papers, deals with the *Differential Equation Method* (DEM), a technique that has proved to be powerful to compute expectation and show concentration for some types of combinatorial structures.

Roughly speaking, the general setting is to have a *discrete random process*, that evolves over time $t = 0, 1, 2, \ldots$, and we are interested in the time development of a

[*] Partially supported by the Spanish CYCIT: TIN2007-66523 (FORMALISM).

[†] Partially supported by the Generalitat de Catalunya through a Beatriu de Pinós grant.

certain property (number of vertices or edges removed in a graph, size of a greedy solution as the algorithm progresses, load issues in balls and bins problems, etc.) Let X_t be a random variable *measuring* the evolution of the property under consideration, at time t. A very basic idea of the DEM is the following: We consider the expected changes between X_t and X_{t+1}, regard the variables as continuous and write the ordinary differential equations suggested by the expected changes. Then we use concentration results to show that as the size of the input grows very large, the random variables are highly concentrated around the value suggested by the solution of the differential equation, with probability tending to 1. That means, the solution of the differential equation is the expected value of the $\{X_t\}$ in the limit, and moreover the distribution of X_t is sharply concentrated around that solution of the ODE. As we will see through many of the examples in this survey, a often use characteristic of the DEM is that the analysis starts with a greedy-type algorithm, without backtracking, to compute a lower bound to the solution to the problem. Then random variables are defined to denote the evolution of a property at each iteration of the algorithm. Kurtz [54] wrote the seminal paper on the use of differential equations to obtain expected value and to prove concentration results in a Markovian discrete process. His formal result is known as *Kurtz's Theorem.* A nice source of the exposition of this result is presented in Chapter 5 of [23]. In posterior papers Kurtz applied his theorem to results on biological and epidemiological models, see for example [55].

One of the first applications of Kurtz's theorem to computer science was made by Karp and Sipser [51]. Denote by $G_{n,p}$ the standard random graph model on n vertices where each edge is chosen independently of all other edges with probability p, see for example [49]. Given a graph, *a matching* is set of edges, in which no two edges have a common vertex. A *maximum matching* is a matching which contains the maximum possible number of edges. A *perfect matching* is a maximum matching which contains all vertices of the graph. Erdős and Renyi [35], studied and estimated the probability of existence of a perfect matching for graphs in $G_{n,p}$, with $p = (\ln n + a)/n - 1$ for constant $a > 0$. Karp and SIpser consider as input graphs $G_{n,p}$ with $p = c/(n-1)$, for constant $c > 0$. Note that the expected number of edges of those graphs is $cn/2$. The authors proposed the following greedy algorithm to find a maximum matching: *Remove all vertices with degree zero, then choose a vertex with minimum degree and select an edge incident with that vertex at random to be in the matching, repeat the process until no new edge can be added.* The authors proved that the execution of the algorithm on $G_{n,c/(n-1)}$ is a Markov chain. Moreover, they used Kurtz's theorem to prove that as $n \to \infty$, with probability tending to 1, the algorithm tends to produce a maximum matching on random graphs of the described type.

Wormald [77] set the appropriate framework to apply the differential equation method in the setting of algorithmic and discrete mathematics. His survey [76] presents a rigorous proof of the general concentration result (Theorem 5.1), which applies to Markovian and non-Markovian processes, together with several examples of application. In the present paper, we aim to do a more intuitive cook-book type survey on *how to apply* the DEM, without entering into the details of the mathematical proof for the general theorem, for which we refer the reader to [76]. Apart from the mentioned survey by Wormald, other surveys have been done about the DEM, some of them for particular types of applications. Among others, we mention the ones by: Zito [79], Achlioptas [2] and Mitzenmacher, Rich and Sitaraman [65]. The survey of Zito deals with the application of the DEM to study the behavior of generic approximation algorithms for some minimization problems on random regular graphs, the survey of Achlioptas deals with the DEM as a tool to obtain lower bounds for the satisfiability of random k-SAT formulae (see Section 5 in the present survey) and the survey of Mitzenmacher et al. presents different techniques for obtaining results on *the power of two random choices* in balls and bins, where one of the techniques described in that survey is the DEM (see also Section 2.1in the present survey).

1.1 Notation and basic definitions-results

Notation. For any $n \in \mathbb{N}$, let $[n]$ denote the set $\{1, 2, \ldots, n\}$. For a graph $G = (V, E)$ we denote by $d(v)$ the degree of vertex $v \in V$, i.e., the number of edges incident to v. Given a vertex v, let $N(v)$ denote the set of neighbors of v. Let K_n denote the *complete graph* on n vertices, and for a graph G, we denote by $\bar{G}$ its complement, i.e. the graph with the same vertex set, and complementary set of edges. Given $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ with $V_1 \cap V_2 = \emptyset$, their *union* $G_1 \cup G_2$ is the graph with vertex set $V_1 \cup V_2$ and edge set $E_1 \cup E_2$. Finally, let δ_{ij} be the Kronecker delta, that is, $\delta_{ij} = 1$ if $i = j$ and 0 otherwise.

For the basic material on probability we refer the reader to the any of the already mentioned basic books or to [44]. Through this paper, u.a.r. will stand for *uniformly at random*, i.e. choosing an element at random from a uniform distribution. The acronym a.a.s. stands for *asymptotically almost surely* and it means with probability $1 - o(1)$, where the asymptotic is respect to the input size, which tends to ∞.

Given a sample space Ω in a probability space, a *σ-field* $\mathcal{A}$ is a nonempty set of subsets of Ω such that:

1. if $A \in \mathcal{A}$ then $\overline{A} \in \mathcal{A}$,
2. the countable union of elements in $\mathcal{A}$ also is in $\mathcal{A}$, and
3. $\emptyset \in \mathcal{A}$.

A continuous or discrete *time random process* $\{X(t) \mid t \geq 0\}$ is said to be *Markovian* if the distribution at time t only depends on $X(t-1)$ and not on the previous history of the system [44]. In the discrete case, a Markovian process is known as a Markov chain. In the next sections we will see several examples of Markov chains. Also, the stochastic process in the aforementioned paper by Karp and Sipser is an example of a Markov chain.

A function $f : \mathbb{R}^d \to \mathbb{R}^d$ is said to be *Lipschitz continuous* if there exists a constant $c > 0$ such that for every $x, y \in \mathbb{R}^d$, we have $|f(x_1) - f(x_2)| \leq c|x_1 - x_2|$. This is a stronger condition than the usual continuity and assures everywhere absolute continuity of the function. A function $f : \mathbb{R}^d \to \mathbb{R}^d$ is Lipschitz continuous if and only if it has bounded first derivative (see for example Chapter 5 of [9]). In Figure 1 we can see a pictorial interpretation taken from that last reference, where the cones are formed by the derivative slopes.

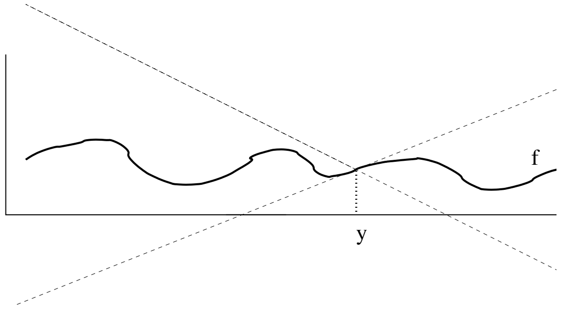

Figure 1: A Lipschitz continuous function $f(y)$. The dotted lines are the cones formed by the derivative slopes.

The basic concentration bounds can be found in any of the books or survey papers already mentioned in the introduction. We briefly explain Hoeffding-Azuma's bound, as it is used in the next section. The reader with basic knowledge in probability theory can skip this part. We skip the proofs, referring the interested reader to one of the mentioned references.

Let us start by introducing the concept of *martingale*. A sequence of random variables $Y_0, \dots, Y_n$ is a *martingale with respect to the sequence* of random variables $X_1, \dots, X_n$, if for all $i \in [n]$, we have $\mathbf{E}\left(Y_i | X_1, \dots, X_{i-1}\right) = Y_{i-1}$. As an example, let $X_1, \dots, X_n$ be a sequence of bounded independent variables with expected value 0. For instance, $X_i = 1$ if the toss of a fair coin results in head and $X_i = -1$ otherwise would be allowed. For any i, $0 \leq i \leq n$, define $S_i = \sum_{k=1}^{i} X_k$. To prove that $S_0, \dots, S_n$ is a martingale with respect to $X_1, \dots, X_n$, we must show that for any i, $\mathbf{E}\left(S_i | X_1, \dots, X_{i-1}\right) = S_{i-1}$.

$$\begin{aligned}\mathbf{E}\left(S_i | X_1, \dots, X_{i-1}\right) &= \mathbf{E}\left(S_{i-1} + X_i | X_1, \dots, X_{i-1}\right) \\ &= \mathbf{E}\left(S_{i-1} | X_1, \dots, X_{i-1}\right) + \mathbf{E}\left(X_i | X_1, \dots, X_{i-1}\right).\end{aligned}$$

Since X_i is independent from $X_1, \dots, X_{i-1}$, we have

$$\mathbf{E}\left(S_i | X_1, \dots, X_{i-1}\right) = \mathbf{E}\left(S_{i-1} | X_1, \dots, X_{i-1}\right) + \mathbf{E}\left(X_i\right) = S_{i-1} + 0.$$

Let $\mathcal{F}_j$ denote the σ-field which contains the collection of all sets which are measurable if the outcomes of the random variables $X_1, \dots, X_j$ are known (the collection of all events which can be assigned probabilities, given $X_1, \dots, X_j$).

In the case of a sequence involving dependent random variables that form a martingale, in order to get concentration results, we need to guarantee that the process does not make big jumps in one step. A martingale $Y_0, \dots, Y_n$ is said to satisfy the *bounded differences condition* if for any $i \in [n]$ there exists a vector $\mathcal{C} = (c_1, \dots, c_n)$ such that $|Y_i - Y_{i-1}| \leq c_i$.

The next result says that when the bounded differences condition holds, the process itself does not go far away from its starting point. It is known as the Hoeffding-Azuma's inequality and the name is due to papers [47] and [10]

Theorem 1 (Hoeffding-Azuma's inequality.)**.** *Let $Y_0, \ldots, Y_n$ be a martingale with respect to $X_1, \ldots, X_n$ that satisfies the bounded differences condition with $\mathcal{C} = (c_1, \ldots, c_n)$. Then, for any $t > 0$,*

$$\mathbf{Pr}\left[|Y_n - Y_0| > t\right] \leq 2e^{\left(-\frac{t^2}{2\sum_{i=1}^{n} c_i^2}\right)}.$$

2 Two simple examples

To introduce the technique, we present two easy examples, which could be solved by other techniques as well. These examples also will yield a gentle introduction to the Differential Equation Method. The first example is to estimate the number of empty balls when we distribute m balls u.a.r. into n bins. The example is inspired from the material of Subsection 4.1.1 in [65]. As a second example, we consider early stages of a random graph process which initially starts with an empty graph and adds random edges, always choosing one vertex of minimum degree, and the example appeared in Wormald's survey [76].

2.1 Balls and bins

Consider the following problem: for some constant $c > 0$ and $n \to \infty$, we are given $m = cn$ balls to be thrown sequentially, independently and u.a.r. into one of n initially empty bins. At every point in the process, we want to determine the number of still empty bins. Denote by $X(i)$, for $i = 0, \ldots, m$ the random variable counting the number of empty bins after i balls have been thrown. Observe that $X(0) = n$ and observe also that the sequence $\{X(i)\}_{i=0}^{M}$ is Markovian, since the number of empty bins at step $i+1$ depends only on the number of empty bins at step i and not on the previous steps. Observe that

$$X(i+1) = X(i) - \delta_{EB}, \tag{1}$$

where δ_{EB} is defined to be 1 if the $(i+1)$-st ball is thrown into an empty bin and 0 otherwise. Since with probability $\frac{X(i)}{n}$ the $(i+1)$'st ball is thrown into one of the $X(i)$ empty bins, and in this case the number of empty bins decreases by one, we have that $\mathbf{Pr}\left[X(i+1) - X(i) = -1\right] = \frac{X(i)}{n}$ and $\mathbf{Pr}\left[X(i+1) = X(i)\right] = 1 - \frac{X(i)}{n}$. Thus

$$\mathbf{E}\left(X(i+1) - X(i)\right) = (-1)\frac{X(i)}{n}. \tag{2}$$

That is, the expected difference of the number of empty bins between throwing two consecutive balls is $\frac{X(i)}{n}$. As the number of bins is always an integer, there is no natural interpretation to convert the *difference equation* given in (2) into an *ordinary* *differential equation* by the canonical approach of taking the limit of two consecutive time steps. This can be accomplished, however, using the following trick: consider a function $x(t)$ defined for all reals with the idea that $x(t)$ models the behavior of $\frac{1}{n}X(tn)$. That is, $x(t)$ represents the scaled number of empty bins when tn balls have been thrown. As initial condition we have $x(0) = 1$. Hence $x(t)$ can be interpreted as a probability of a random bin to be empty after tn balls. Then, the difference equation (1) becomes

$$x(t + \frac{1}{n}) - x(t) = -\delta_{EB}. \tag{3}$$

As $n \to \infty$, we can take the limit of (3), and presuming that the changes in the functions are equal to their expected changes, we obtain

$$dx/dt = -x. \tag{4}$$

Note that up to now, the differential equation is only *suggested*, as the steps of the process are not completely independent, but in Section 3 we show that indeed this is the right solution. We can solve this equation by separation of variables and obtain $-dx/x = dt$, which after integration yields $x(t) = e^{-t}C$ for some constant C. From the initial condition $x(0) = 1$ we obtain that $C = 1$. In plain words, at time t, when tn balls are thrown, the expected fraction of empty bins is e^{-t}. Moreover, after having thrown any number T of balls, the number of empty bins is concentrated around its expected value. To prove the concentration, we use a direct approach using Azuma's inequality (in Section 3 we use a different approach). Denote by $\mathcal{F}_j$ the σ-field of events corresponding to the possible states after j balls have been thrown, for $0 \leq j \leq cn$, and define by $Z_j = \mathbf{E}\left(X(T) | \mathcal{F}_j\right)$ be the associated conditional expectation of $X(t)$. The random variables $\{Z_j\}_{j=0}^{T}$ form a martingale sequence, and moreover, $|Z_j - Z_{j-1}| \leq 1$. Thus, by Azuma's inequality, $\mathbf{Pr}\left[|X(T) - \mathbf{E}\left(X(T)\right)| > \sqrt{\alpha T}\right] < 2e^{-\alpha/2}$.

2.2 The first phase in the min-degree-graph process

The min-degree graph process is the following random graph process: start with an initially empty graph $G_0 = \bar{K_n}$, where n is assumed to be large. For $t > 0$, given G_t choose u.a.r. a vertex u of minimum degree in G_t, then choose u.a.r. a vertex v not adjacent to u in G_t. Set G_{t+1} to be the graph whose edges are the union of the edges of G_t and $\{uv\}$. The process finishes with the complete graph.

In order to study the structure of the evolving graph G_t, a first step is to find the expected number of vertices of a given degree. It turns out to be helpful to split the process into various phases: the process is in phase k, if the minimum degree of the graph G_t is k. That is, if $Y_i(t)$ denotes the number of vertices of degree i in G_t, k is the minimum number such that $Y_k(t) > 0$. Let us consider phase 0 in detail: for every edge uv to be added, vertex u has degree 0, and with probability $\frac{Y_i(t)}{n-1}$, the second vertex v has degree i, since v can be any vertex except for u. Observe that in phase 0 the expected change of the values $Y_i(t)$ is completely determined by the values of $Y_i(t-1)$ and does not depend on the previous history. In other words, the process is Markovian. Note that in phase 0, for one fixed time step t, $Y_0(t)$ decreases by 1 or by 2, depending on whether $d(v) = 0$. Similarly, $Y_1(t)$ either increases by two (if $d(v) = 0$), it increases by one (if $(d(v) \geq 2)$, or it stays the same (if $d(v) = 1$). For $i \geq 2$, $Y_i(t)$ might increase or decrease or remain unchanged (depending on $d(v)$). Thus, defining X_i to be the indicator variable that $d(v) = i$, we have the following difference equations:

$$\begin{aligned} Y_0(t+1) &= Y_0(t) - 1 - X_0, \\ Y_1(t+1) &= Y_1(t) + 1 + X_0 - X_1, \\ Y_i(t+1) &= Y_i(t) + X_{i-1} - X_i, \quad i \geq 2. \end{aligned} \tag{5}$$

After taking expectations, we get

$$\mathbf{E}\left(Y_i(t+1) - Y_i(t) | G_t\right) = -\delta_{i0} + \delta_{i1} + \frac{Y_{i-1}(t) - Y_i(t)}{n-1}, \quad i \geq 2, \tag{6}$$

As before, in order to pass to the limit, we consider a function $z_i(x)$ in order to model the behavior of $\frac{1}{n}Y_i(xn)$. As before, presuming that the changes in the functions Y_i are equal to their expected changes, we obtain

$$z_i'(x) = -\delta_{i0} + \delta_{i1} + z_{i-1}(x) - z_i(x), \quad i \geq 0, \tag{7}$$

where $z_{-1}(t) = 0$ for all t. Since at time 0 there are only vertices of degree 0, we have $z_0(0) = 1$ and $z_i(0) = 0$ for all $i > 0$ as initial conditions. These differential equations are solved iteratively. For $i = 0$ the solution of $z_0'(x) = -1 - z_0(x)$ can be obtained as before, by separation of variables we obtain $\int \frac{1}{-1-z_0} dz_0 = \int dx$, and the general solution is $z_0(x) = e^{-xc} + 1$. Taking into account the initial condition $z_0(0) = 1$, we see that $z_0(x) = 2e^{-x} + 1$. Solving for $z_1(x)$ we obtain $z_1(x) = 2xe^{-x}$ (using the initial condition $z_1(0) = 0$) and in general we get $z_i(x) = \frac{2x^i}{i! e^x}$. As before, for any step T in phase 0 and for any fixed i one can show concentration using Azuma's inequality: For any $j = 0, \ldots, T$, define $\mathcal{F}_j$ to be the σ-field of events corresponding to the possible states after j edges have been added, and define the random variable $Z_j^i = \mathbf{E}\left(Y_i(j) | \mathcal{F}_j\right)$. Then, $\{Z_j^i\}_{j=0}^T$ forms a martingale, and $|Z_j^i - Z_{j+1}^i| \leq 1$. Apply again Azuma's inequality to obtain

$$\mathbf{Pr}\left[|Y_i(T) - \mathbf{E}\left(Y_i(T)\right)| > \sqrt{\alpha T} \right] < 2e^{-\alpha/2}.$$

In the next section we will present a basic result of Wormald [77] which shows that the suggested solution is indeed the unique solution and at the same time gives concentration, hence making the application of Azuma's inequality unnecessary.

Later phases of the min-degree random process can be analyzed in a similar way. One writes down the new differential equations, and the initial conditions are determined from the previous phases. However, there is a technical detail which complicates the analysis. If the process is in phase $k > 0$, then the chosen vertex u is already adjacent to some other vertex, and hence such a vertex cannot be chosen as second

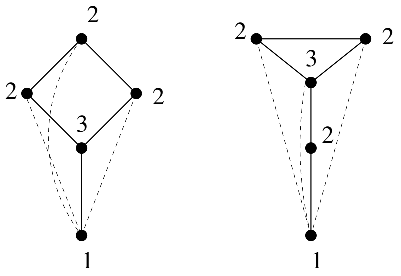

Figure 2: Two graphs in phase 1 with identical degree distributions but different transition probabilities

vertex anymore. Consider the example in Figure 2 where two graphs with the same $Y_i(t)$ values are given, but the probability that the second vertex to be added has degree 2 is not the same in both cases (in the first case it is 1, and in the second case it is $\frac{2}{3}$). That is, the values $Y_i(t+1)$ do not only depend on $Y_i(t)$, but also on previous steps, and hence the process is no longer Markovian. It can be shown, however, that as long as the graph does not contain too many edges, the already existing edges are rare and do not affect the expected values by too much.

3 The basic element of the analysis: Wormald's theorem

The goal of this section is to give a rather simple and intuitive explanation of the basic element of the analysis of algorithms using differential equations. The objective is to give a theorem which is simple to apply, without aiming for maximum generality; for a more general setup we refer the reader to [76]. In a nutshell and very roughly speaking, Wormald's theorem [75, 76, 78] states the following: We are given a dynamic process where the expected changes of some property P at every step are small, and whose expected changes are equal to some "smooth", differentiable functions. When modeling the changes by a system of these differential equations, the system has a unique solution determined by the initial value conditions; moreover, the behavior of the property P is at every moment sharply concentrated around the (properly scaled) solution of the differential equations. The setting of the ordinary differential, such that they agree with the hypothesis in Theorem 2, is what we denote by the differential equation method (DEM). The core of the method is the following simplified version of Wormald's theorem (see also [78]),

Theorem 2. *Given random variables Y_i, for $1 \leq i \leq a$ (where $a > 0$ is a constant), representing components of a time discrete Markov process $\{G_t\}_{t \geq 0}$. Assume that $D \subseteq \mathbb{R}^{a+1}$ is closed and bounded and contains the set*

$$\{(0, y_1, \ldots, y_a) : \mathbf{Pr}[Y_i(0) = y_i n, 1 \leq i \leq a] \neq 0 \textit{ for some } n\}.$$

In words, D has to contain all points corresponding to initial conditions. If for all i and all t, $|Y_i(t+1)-Y_i(t)| \leq \beta$ for some constant $\beta > 0$, and $|\mathbf{E}\left(Y_i(t+1) - Y_i(t)|G_t\right) - f_i(t/n, Y_1(t)/n, \ldots, Y_a(t)/n| \leq \lambda$ for some $\lambda = o(1)$ and some functions f_i which are Lipschitz continuous on an open set containing D, then, for $(0, \hat{z_1}, \ldots, \hat{z_a}) \in D$ the system of differential equations

$$\frac{dz_i}{d_x} = f_i(x, z_1, \ldots, z_a)$$

has a unique solution for $z_i : \mathbb{R} \to \mathbb{R}$ in D passing through $z_i(0) = \hat{z_i}$. Moreover, asymptotically almost surely

$$Y_i(t) = nz_i(t/n) + o(n),$$

uniformly for all t and all i, where $z_i(0) = \hat{z_i} = \frac{1}{n}Y_i(0)$.

Remark.

- The theorem also holds when a is a function of n, and when D is a function of n provided that all functions f_ℓ are uniformly bounded by some Lipschitz constant and if f_ℓ depends only on the variables $x, y_1, \ldots, y_\ell$.
- In fact, both conditions have to hold only for $t < T_D$, where T_D is the minimum t such that $(t/n, Y_1(t)/n, \ldots, Y_a(t)/n) \notin D$.
- The last sentence of the theorem is in fact true only for $t \leq \min\{\sigma n, T_D\}$, where $\sigma = \sigma(n)$ is the supremum of those x to which the solution can be extended before reaching the boundary (or getting very close to the boundary) of D. The $o(n)$ term can be made more precise, see again [76] for details.
- The theorem also holds for non-Markovian processes if certain conditions on the history of the process are satisfied, see once more [76] for details.

We will not give a proof of the theorem, which can be found either in [78] or in [76]. Instead, we sketch the main ideas of the proof. The expected changes suggest a *mean path*, that is, a path, the random variables would follow if in each step the actual change was exactly equal to the expected change. In general, arbitrary random variables do not have to remain close to their mean path, but in a setting, where the expectation of the change is a smooth function, and where each step has only small effects, the probability, that the random variables deviate significantly from their conditional expectation, is small. Actually, having small effects with high probability would also be sufficient, i.e., the first condition does not necessarily have to hold always (see [76] for details).

The reason for considering scaled versions is to capture the notion of smoothness of random variables. The uniqueness of the solution follows by a standard result in the theory of first order differential equations. Once again we point out that the theorem is true in much more general form, and we refer again to [76] and the references therein for details. We remark that the big advantage of Wormald's theorem over the classical method of applying Azuma's inequality in the analysis of algorithms is that the theorem not only gives concentration, it also provides the expected value around which the solution is concentrated.

As a first application, we apply Theorem 2 to prove uniqueness and concentration for the example given in Subsection 2.1. The values $X(t)$ play the role of $Y_i(t)$: The sequence $\{X(t)\}_{t=0}^{m}$ is Markovian and thus $x(t)$ is Markovian as well. Since $x(0) = 1$, the point $(0, 1)$ has to be an element of D. One can choose $\beta = 1$ and $f(t, x) = -x$, which is Lipschitz continuous for all values of $-1 < t < 2c$ and for $x > 0$. Thus the differential equation $\frac{dx}{dt} = -x$ has as unique solution passing through $x(0) = 1$, namely the solution $x(t) = e^{-t}$. Moreover, $X(t)$ is concentrated around $ne^{-t/n}$.

In a similar way, we can apply Theorem 2 to the example given in Subsection 2.2: We have $a = n$, and the theorem applies as well. We can set $f_i = -\delta_{i0} + \delta_{i1} + z_{i-1} - z_i$ (recall that δ_{ij} is 1 for $i = j$ and 0 otherwise). For example, D can be defined by $z_0 > 0$ and $-1 < x < 3$ and $-1 < z_i < 2$ for $i > 0$. Then the theorem can be applied with $\lambda = \max \frac{|Y_{i-1}(t)-Y_i(t)|}{n(n-1)} \leq \frac{1}{n-1}$ and $\beta = 2$.

4 Problems on random r-regular graphs

A *r-regular graph* is a graph where each vertex has exactly r neighbors. Due to the particularly nice spectral and combinatorial properties of regular graphs, a large body of literature have been developed studying properties of both the deterministic and random versions of those graphs, see for example the survey [76].

The DEM has been shown to be an efficient tool to analyze algorithms and to find expected size of combinatorial properties in *random r-regular graphs*. In this section, we will sketch some of the work done on random r-regular graphs, where the DEM has been used as one of the main techniques.

4.1 Generation of random r-regular graphs

Let $\mathcal{G}_{n,r}$ denote the probability space of r-regular graphs with n vertices and degree $3 \leq r < n$, where we shall assume that nr is even. To *sample* a graph u.a.r. from $\mathcal{G}_{n,r}$ is to choose a graph with probability $1/|\mathcal{G}_{n,r}|$, among the space of all *simple* r-regular graphs with n nodes, where simple means without self-loops or multiple edges. To design efficient algorithms to implement the sampling, Bender and Canfield [11] and Bollobás [13] introduced the *pairing model*, also referred to as the *configuration model* in [13]. The pairing model is well described in several places, among others in Chapter 9 of [49], Section 7.6 of [15] and Section 2 in [76].

Section 2.1 in the survey [77] gives a detailed account of the genesis of the pairing model. Here we just mention the main events, with an update of the recent developments. The original configuration model from Bender-Canfield and Bollobás, which is given in Algorithm 1, is easy to apply and gives a.a.s. simple graphs drawn u.a.r., in the case when r is a constant or r grows slowly as $r = O(\log n)$, where as usual n is the size of the graph. From the practical point of view, the algorithm is quite slow.

McKay and Wormald [61] gave a $O(nr^3)$ algorithm, which is difficult to implement, moreover it works for values of $r = o(n^{1/2})$. Later, Steger and Wormald [73], gave an $O(nr^2)$, easy to implement algorithm algorithm to generate r-regular graphs u.a.r., for values up to $r = o(n^{1/3})$.

For our purposes, it will be enough to consider the following procedure to generate *simple* graphs in the *pairing model*: given n vertices $\{v_1, \ldots, v_n\}$, let N a set of nr points, r points in each vertex, for r constant.

Algorithm 1 Algorithm for obtaining a simple random r-regular graph

```
given $n$ vertices $\{v_1, \ldots, v_n\}$ and $r$
while there is an empty pair of points repeat
    select u.a.r. two empty points and expose them
      by joining them with an edge
    if the new edge is a loop within $v_i$ or
      forms a multi-edge between $v_i$ and $v_j$
      then stop and start again the whole process
output the $r$-regular graph with vertex set $\{v_1, \ldots, v_n\}$
```

As explained in detail at the beginning of Section 2 in [77], asymptotically in n, Algorithm 1 selects u.a.r. a simple random r-regular graph in $\mathcal{G}_{n,r}$, with probability $e^{(1-r^2)/4}$.

In the left of Figure 3, we have a toy example of a 4-regular random graph with $n = 10$. The drawing to the right is the representation of the graph, where each vertex is a bin with four points, together with a matching between points in different vertices, which generate a simple uniform graph. Algorithm 1 only works efficiently for very large values of n.

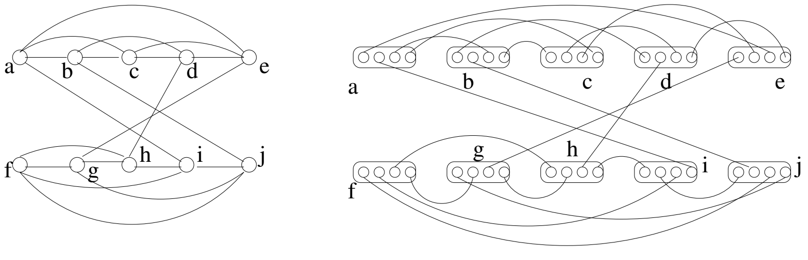

Figure 3: A graph in $\mathcal{G}_{10,4}$ and the matching producing it.

4.2 The independence number of a r-regular graph

Recall that given a graph, a set of vertices in the graph is *independent* if no two vertices in the set share an edge. Finding a *Maximum Independent Set* (MIS) on a graph, even on r-regular graphs, is NP-hard [42]. Given a graph G its *independence number* $\alpha(G)$ is the size of its MIS, i.e. if S denotes a MIS, then $\alpha(G) = |S|$.

For $G \in \mathcal{G}_{n,r}$, Frieze and Łuczak [39] proved that

$$\mathbf{Pr}\left[|\alpha(G) - \frac{2n}{r}(\log r - \log\log r + 0.306)| \leq \epsilon/r\right] \to 1, \text{ as } n \to \infty,$$

for r fixed (depending on ϵ) but arbitrary large. So their results do not say anything about values of $r = 3, 4, \ldots$.

When the DEM is applied to bound maximum (minimum) size of a set of vertices (edges, cycles, matchings,..) with a given property, for a random r-regular graph, one way to go is to combine the greedy algorithm to obtain the solution, with the progressive exposing of the edges in the graph $G \in \mathcal{G}_{n,r}$.

4.2.1 Greedy algorithm to approximate the independence number of a $\mathcal{G}_{n,r}$

We present a first algorithm to approximate the independence number of a random r-regular graph G, which is taken from [75]. As it is customary when using the DEM to perform the analysis, the graph is *exposed* at the same time the algorithm builds the solution set S for the MIS. The algorithm starts with a set $Z = V$ of n vertices. At iteration $t = 0$, Z consists of all vertices with degree 0. At each iteration of the algorithm, we increment S by one, and expose the edges from that vertex, removing from Z the vertices with degree different from 0. The algorithm finishes when $Z = \emptyset$. Figures 4 to 7 represent the evolution of the algorithm on the random 4-regular graph from Figure 3. In the left, the evolution of the graph G_t as we expose edges from the points in the vertex selected to be part of the solution S. The vertices selected to be part of the solution are shown with a thick circle, the neighbors discarded are crossed out. In the right the same graph but with the vertices as bins, each with 4 points, which, when paired, produce the edges of the graph. Again the reader should take this toy example with prevention, as the analytical results obtained from the applications of the DEM work only asymptotically. For instance, if with probability 2/10 we started by vertices a or j, the resulting independence number would have been 2.

Algorithm 2 Greedy algorithm for approximating $\alpha(G_{n,r})$

```
given $V = \{v_1, \ldots, v_n\}$ vertices from a $\mathcal{G}_{n,r}$ and $r$
let $S = \emptyset$ and $Z = V$
while $Z \neq \emptyset$ repeat
    choose u.a.r. a vertex $v \in V$ and add it to $S$
    expose all the $r$ edges from $v$ to other vertices
    remove from $Z$: $v$ and all $w$ s.t. $vw$ has been exposed
output $\alpha = |S|$
```

Algorithm 2 runs for a number of steps between 0 and t_f, where $t_f = |S|$. Notice that at t_f the zero-degree vertices disappear, i.e., $Z = \emptyset$. Let Y_t be the random

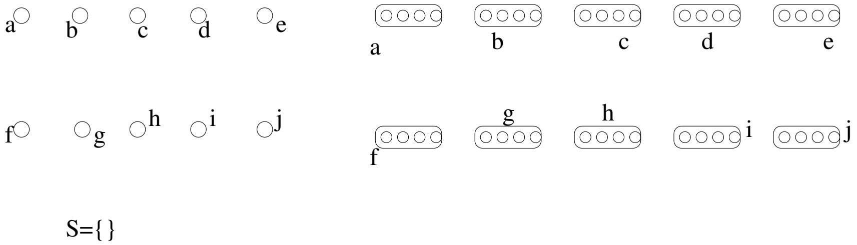

Figure 4: Initial situation of the algorithm for MIS in graph in $\mathcal{G}_{10,4}$.

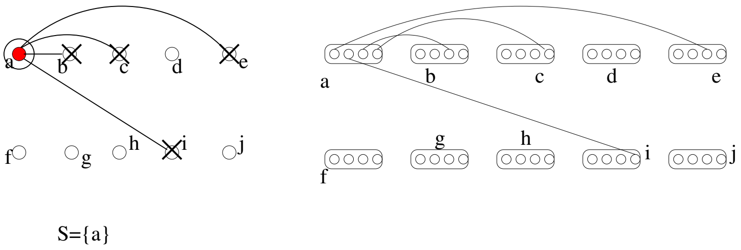

Figure 5: First iteration of the algorithm for MIS in graph in $\mathcal{G}_{10,4}$ and the edges exposed.

variable representing the number of zero-degree vertices, then

$$|Y_{t+1} - Y_t| \leq r + 1. \tag{8}$$

Notice that when $Y_{t_f} = 0$, the S will be the output for MIS and $\alpha(G) = |S|$. The aim is to get a lower bound on the expected size of the set $\alpha(G)$.

At any iteration t, after choosing a new vertex in Z and incorporate it to S we expose all the edges out of it, so the number of *free* (unmatched) points is $nr - 2tr$. Therefore the probability that a randomly selected point belongs to a vertex in Z is

$$\frac{rY_t}{nr - 2rt} = \frac{Y_t}{n - 2t}.$$

The probability changes slightly, when a new vertex of degree 0 is chosen, but as long as $2t + 2 < n$, for each of the r edges exposed in this step, the probability of joining

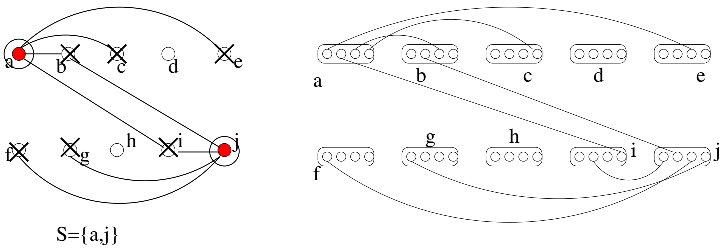

Figure 6: Second iteration of the algorithm for MIS in graph in $\mathcal{G}_{10,4}$ and the edges exposed.

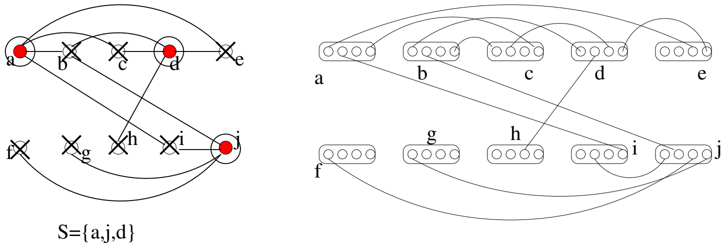

Figure 7: A graph in $\mathcal{G}_{10,4}$ and the edges exposed.

to a vertex in Z is still $\frac{Y_t}{n-2t} + O(1/n)$. Therefore, assuming $2t + 2 < n$,

$$\mathbf{E}\left(Y_{t+1} - Y_t \,|\, G_t\right) = -1 - \frac{rY_t}{n-2t} + O(\frac{1}{n}).$$

Notice that $\{G_t\}_{t=0}^{t_f}$ is a Markovian process.

The above equation implies that the expected decrease of Y_t is $-1 - \frac{rY_t}{n-2t}$. Taking $Y_t = nz(x)$ with $x = t/n$, we define the function

$$f(x,z) = -1 - \frac{dz(x)}{1-2x}.$$

Define also the domain in $\mathbb{R}^2$, $D = \{((x,z) \,|\, -\epsilon < x < 1/2, \epsilon < z < 1-2x\}$. for small $\epsilon > 0$. Then we have the first order differential equation

$$\frac{dz(x)}{dx} = f(x,z),$$

with initial condition $z(0) = 1$, which has solution

$$z(x) = \frac{(r-1)(1-2x)^{r/2} - (1-2x)}{r-2}.$$

As $z(x)$ indicates the evolution of Y_t, and the solution is obtained when $Y_t = 0$, then the first positive solution of $z(x) = 0$ is

$$\frac{1}{2}(1 - (\frac{1}{r-1})^{2/(r-2)}). \tag{9}$$

It is straightforward to see that the conditions in Theorem 2 are fulfilled and therefore the expression in (9) is the expected value for $\alpha(G)$ obtained by Algorithm 2, and the value obtained is concentrated.

4.2.2 A better bound on the independence number of random r-regular graphs.

Wormald in [75] improved the previous bound using an algorithm based on choosing the maximum degree vertex at the same time as exposing the graph, see also Section 7.1 in [76].

In the new algorithm, as before, the only given information is the vertex degree $r \in \Theta(1)$ with $r \geq 3$ and the number n of vertices. Let V denote the vertex set in the current iteration of the algorithm. At the first step the algorithm chooses one vertex v u.a.r. from the set of all possible vertices. Expose all its edges, mark its neighbors and expose all the edges going out of the marked neighbors. This process yields a graph G_1 whose vertex set V consists of v plus all marked neighbors of v plus the neighbors of the marked vertices and the edge set consists of all exposed edges. Notice that the degree of the neighbors of $N(v)$ could be strictly less than r. From there on, at each iteration, one keeps a set M of the maximum degree vertices in V, which are non-marked and do not form part of the current independent set S. The formal description is given in Algorithm 3, where the variable b is a boolean variable used to differentiate the first case from the remaining ones. As before, the goal is to obtain a lower bound for the independence number $\alpha(G) = |S|$, for $G \in \mathcal{G}_{n,r}$. Notice that at the end of the execution of the algorithm all the edges must be present. In Figure 8, we show the two first steps of the algorithm on a set of 10 vertices that at the end yields the independence number of the random 4-regular graph in Figure 3. On the left the graph with the n possible vertices (vertices are represented in dotted line until incorporated to V). On the right, the result of the first iteration. The vertex a is incorporated into the solution S. V immediately becomes the set of all vertices, $\{b, c, e, i\}$ are marked, $M = \{d\}$. At the left in Figure 9 a representation of the second iteration, where the algorithm chooses d into S and $M = \{f, g, j\}$. Finally i the right picture represents the resulting graph, after adding j.

Algorithm 3 Minimum degree greedy algorithm for obtaining a lower bound on the independence number of $\mathcal{G}_{n,r}$

```
given $n$ and $r$
let $S = \emptyset$ and $M = \emptyset$, $b = 0$
while $M \neq \emptyset$ or $b = 0$ repeat
  if $b = 0$, choose $i$ u.a.r. between 1 and $n$ set $b = 1$ and $v = v_i$
    else choose $v$ u.a.r. from $M$
  expose all the $r$ edges from $v$ to $r$ vertices (will be $N(v)$) and mark all $w_i \in N(v)$
  expose all the $r$ edges from each $w_i \in N(v)$ to their neighbors
  add $v$ and all marked $w_i$ and the neighbors of each $w_i$ to $V$
  $S := S \cup \{v\}$
  $M$ = set of maximum degree vertices in $V - (S \cup \{\text{set of marked vertices}\})$
output $\alpha = |S|$
```

We sketch the proof given in [75], for the full details the reader should read the paper by Wormald (see also Section 7.1 in [76]). During the execution of the algorithm, define a vertex to be *unsaturated* if its degree still is $< r$. Assume n is very large. The degree of the chosen vertex to be incorporated to S affects the final size $|S|$, moreover as the execution of Algorithm 3 progresses, the minimal and maximal degree of the unsaturated vertices evolves. For instance, at the first iterations of Algorithm 3, we have a large amount of vertices with degree 1, but when $|S| > cn$, for small $c > 0$,

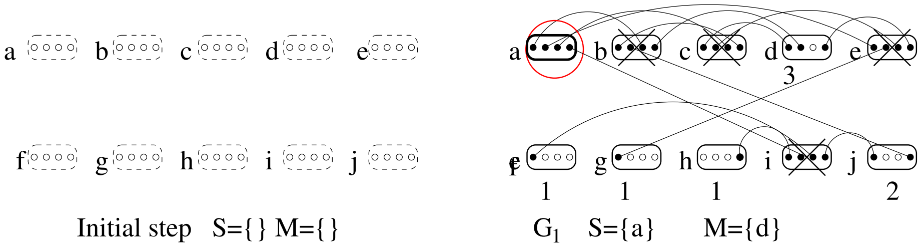

Figure 8: Initial situation of the algorithm for MIS in graph in $\mathcal{G}_{10,4}$ and the matching producing it.

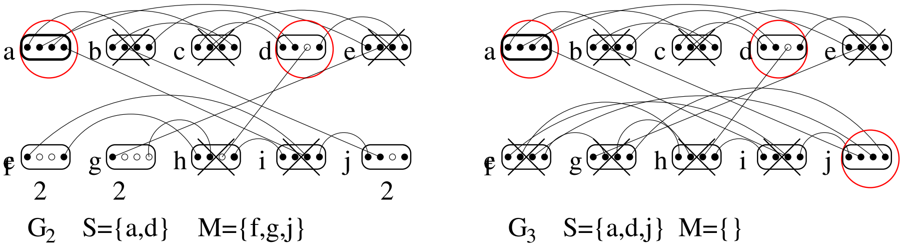

Figure 9: First iteration of the algorithm for MIS in graph in $\mathcal{G}_{10,4}$ and the matching producing it.

vertices with degree 2 or higher begin to appear. In general, if at a certain moment in the execution of the algorithm, most of the unsaturated vertices are of degree k or $k+1$, we say that the execution of the algorithm is in phase k. In [75] it is proved that each phase lasts $o(n)$ additions of vertices to S. Notice that phases will go until phase $r-2$ a.a.s. (the appearance of phase $r-1$ is possible, but it can be shown that a.a.s. this does not occur).

Suppose now that we are in phase k. Define a *clutch* to be a set of vertices added to S between the addition of a vertex of degree k and the next addition of a vertex of degree k, i.e. at some iteration when incorporating one vertex of degree k to S, all the remaining unsaturated vertices may have degree $< k$, so it will take a number of iterations to produce another unsaturated vertex of degree k. Consider every clutch as a unit of time. For each phase k, define G_t as the graph "exposed" (vertices considered with the exposed edges) after t clutches. Let $Y_i(t)$ the number of vertices with degree i in G_t and let $U(t)$ the number of unpaired points remaining at time t. Then,

$$U(t) = \sum_{i=0}^{r-1} (r-i) Y_i(t).$$

To compute the increment of the number of vertices of degree i at G_t, if v with $d(v) = k$ is incorporated to S, notice that when a point in v is chosen at random to be paired with a free point outside of v, the probability that the new point lies in a vertex of degree j is $(r-j)\frac{Y_j(t)}{U(t)}$ (see Figure 10). Moreover, the expected number of vertices of degree j matched by the exposition of the free points in vertex v incorporated to S

is $\frac{(r-k)(r-j)Y_j(t)}{U(t)}$. As each one of the neighbors of v also expose theirs free points, the expected number of vertices of degree i that will have a point matched to one point in the neighbors of v is

$$\frac{(r-k)(r-j)Y_j(t)}{U(t)} \times \frac{(r-j-1)(r-i)Y_i(t)}{U(t)}.$$

As the vertices in $N(v)$ change from degree j to r, they contribute to a reduction of Y_j. The vertices which are neighbors to the vertices in $N(v)$ change from degree i to degree $i+1$, so they increase Y_{i+1} and decrease Y_i. If we add all this over all $r-2$ phases, we get that when we expose edges due to incorporating a vertex v with $d(v) = k$ to S, the expected increase of Y_i is

$$-\delta_{ik} + (r-k)\mu_i, \tag{10}$$

where δ_{ik} is the delta of Kronecker and

$$\mu_i = -\frac{(r-i)Y_i(t)}{U(t)} + \sum_{j=0}^{r-2} \frac{(r-j-1)(r-j)((r+1-i)Y_{i-1}(t) - (r-1)Y_i(t))Y_j(t)}{U(t)^2}.$$

On the other hand, if v has degree $k+1$ the expected change in Y_i (for $i \leq k$) is

$$(r-k-1)\mu_i. \tag{11}$$

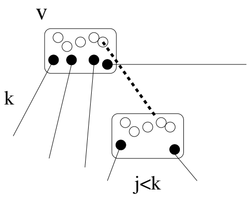

Figure 10: One vertex v of degree k is added to S, and exposes the remaining $r-k$ points, one of them is matched to a point of degree $j < k$.

To find the expected number of vertices of degree $k+1$ in a clutch, we begin with $Y_{k+1}(t) = 0$. When in a new clutch the first vertex is incorporated to S and the algorithm exposes all edges in the vertex and the edges of the neighbors, by (10) the expected number of vertices with degree $k+1$ in the clutch is $(r-k)\mu_{k+1}$. The evolution of the vertices of degree $k+1$ in the k-th clutch follows a *birth/death process*, where births represent the new vertices of degree $k+1$ being created, and a death represents the disappearance of one such vertex. Then, the expected number of births in the whole birth/death process is

$$\frac{(r-k)\mu_{k+1}}{1-(r-k-1)\mu_{k+1}}. \tag{12}$$

From (10), (11) and (12) we get that for $i \leq k$, conditioning on G_t, the expected changes in Y_i due to the k-th clutch is

$$\mathbf{E}\left(Y_i(t+1) - Y_i(t) \,|\, G_t\right) = \frac{(r-k)\mu_{k+1}}{1-(r-k-1)\mu_{k+1}} - \delta_{ik}.$$

This equation suggests the following system of differential equations. If xn represents the number of clutches in the whole process (i.e., $x = t/n$), then, writing $Y_i(t) = nz_i(x)$, $\mu_i(t) = n\tau_i(x)$, $U(t) = \gamma(x)$, one obtains the system

$$\begin{aligned} \frac{dz(x)}{dx} &= \frac{(r-k)\tau_i(x)}{1-(r-k-1)\tau_{k+1}} - \delta_{ik} && \text{if}(i \leq k), \\ z_i(x) &= 0 && \text{otherwise,} \end{aligned}$$

where

$$\tau_i = -\frac{(r-i)z_i}{\gamma} + \sum_{j=0}^{r-2} \frac{(r-j-1)(r-j)((r+1-i)z_{i-1} - (r-i)z_i)z_j}{\gamma^2}$$

and

$$\gamma = \sum_{i=0}^{r-1} (r-i)z_i.$$

Manipulating a bit the system, it is easy to see that the conditions of Wormald's theorem are fulfilled, so the solution to the ODE will yield the lower bound on the independence number of the random r-regular graph (again for details see [75]).

For random 3-regular graphs, the system of ODE can be explicitly computed to give a lower bound on $\alpha(G)$ of $6\ln\frac{6}{3}n = 0.4328n$. This result obtained by the algorithm 3 using the DEM coincides for the value $r = 3$ with the result obtained by Frieze and Suen [40], using martingale theory. We feel that in this case the DEM is simpler to apply. Recall that in the previous example using Algorithm 2, the lower bound obtained for random 3-regular was $0.3750n$. For other values of r, the system of ODE can be solved numerically. For example, the lower bound for α in random 4-regular graphs is $0.3901n$.

Regarding upper bounds on the independence number of a random r-regular graph generated by the pairing model, Bollobás [15] gives a general formula. McKay [60] gave more accurate numerical values, for some small constant r. For those values of r, it was shown in [75] that the lower bounds obtained by Algorithm 3 are quite close to the upper bounds obtained by McKay. For instance, McKay's upper bound for 3-regular is $0.4554n$, while for 4-regular graphs is $0.4163n$. Later, Colin, Frieze, Reed and Riordan [22] proved that for any arbitrarily small constant $\epsilon > 0$, if $3 \leq r \leq n^{1-\epsilon}$, if G is chosen u.a.r. from the set $\mathcal{G}_{n,r}$ $(n \to \infty)$, $\alpha(G)$ is asymptotically equal to $\frac{2n\log r}{r}$.

4.3 Bisection width of a random 4-regular graph

Given a graph $G = (V, E)$ with $|V|$ even, a *bisection* of G is a partition of V into two equal size parts. The *size* of the bisection is the number of edges crossing between the parts. The *bisection width* (or minimum bisection of G) is the value of the minimum size bisection in G. In the same way we can define the *maximum bisection* as the maximum size bisection. For the graph in Figure 11 we get a bisection width 1 and maximum bisection 5.

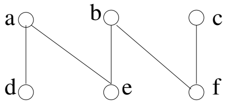

Figure 11: A graph with minimum bisection 1 and maximum bisection 5.

The problems of finding the bisection width and the maximum bisection are NP-hard for r-regular graphs, for any $r \geq 3$ [42]. Regarding previous work on the bisection size, Bollobás provided a lower bound of $(r - 2\sqrt{r \ln 2})n/4$, for almost all r-regular graphs [14]. In the same paper, he gave a lower bound of $0.22n$ for the particular case of 4-regular graphs. In [53] it is shown that almost all cubic graphs have bisection width greater than $0.101n$. Recently, using spectral techniques, Bezroukov, Elsässer, Monien, Preis and Tillich produced lower bounds of $0.082n$ for the bisection width of cubic Ramanujan graphs, and of $0.176n$ for the case of 4-regular Ramanujan graphs [12]. In [25] the authors present tight upper and lower bounds for the expected bisection width and the expected maximum bisection on random 3-regular and random 4-regular graphs. In this subsection, we present the application of the DEM to the bisection width of random 4-regular graphs.

Given a graph in $\mathcal{G}_{n,r}$, and given a partial assignment of colors red (R) and blue (B) to its vertices, we classify the non-colored vertices according with the number of their colored neighbors: A vertex is of *Type* (r, b) if it has r neighbors colored R and b neighbors colored B. A pair of uncolored vertices is said to be (r, b)-*symmetric* if their types are (r, b) and (b, r) (for the same values of b and r).

The greedy procedures work by coloring vertices chosen randomly in symmetric pairs, to maintain balance, and repeatedly use the following operations: the *majority operation* (Maj), that colors each vertex with the majority color among its neighbors (see Figure 12).

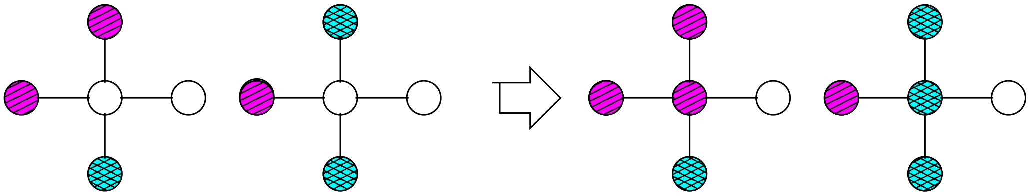

Figure 12: The Majority operation with a symmetric pair $(2, 1)$ and $(1, 2)$.

The algorithm 4-min greedy considers only (0,1)-, (0,2)- and (1,2)-symmetric pairs of uncolored vertices and gives higher priority to the (0,2)- and (1,2)-symmetric pairs than to the (0,1)-symmetric pairs. Note that the size of the bisection is the number of *bicolored* edges, with one vertex of each color, so only each Maj operation on a (1,2)-symmetric pair contributes, with 2, to the bisection.

Algorithm 4 Algorithm 4-min greedy for obtaining a bisection of a $\mathcal{G}_{n,4}$ graph

```
Given a random 4-regular graph $G$
select two non-adjacent vertices u.a.r., color one R and the other B
Phase 1: repeat
  if there are vertices of both types (2,0) and (0,2)
    or vertices of types (2,1) and (1,2)
  then
    select u.a.r. a (0,2)- or (1,2)-symmetric pair and perform Maj
  else if there are vertices of both types (1,0) and (0,1)
  then
    select u.a.r. a (0,1)-symmetric pair and perform Maj
until no new vertex is colored
Phase 2: color randomly the remaining uncolored vertices, half R and half B
  output the bisection R, B.
```

In Figures 13 to 15, there is the simulation of Algorithm 4 on the toy graph $G_{10,4}$ given in previous examples. The light color represents B and the darker color represents R. In the last step, the algorithm assigns the colors randomly.

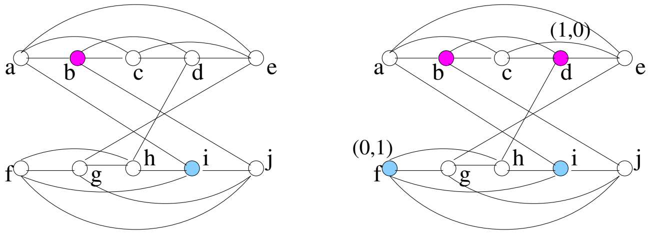

Figure 13: Steps 1 and 2 of the algorithm on $G_{10,4}$.

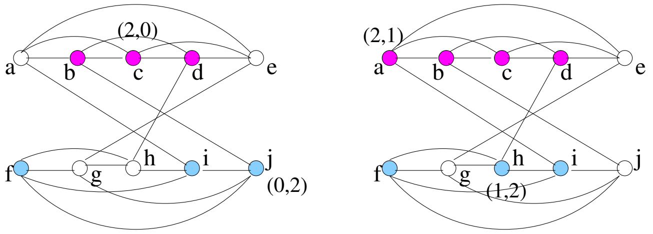

Figure 14: Steps 3 and 4 of the algorithm on $G_{10,4}$.

Let Z_{rb} represent the number of uncolored vertices of type (r, b), and let W denote the number of points not yet involved in exposed pairs. Then,

$$W = \sum_{r+b \leq 4} (4 - r - b) Z_{rb}.$$

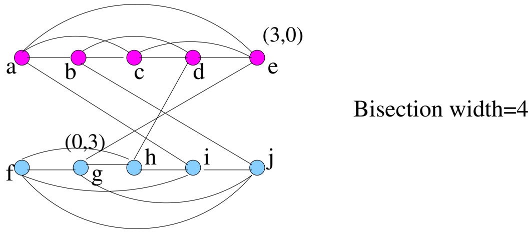

Figure 15: Step 5 of the algorithm on $G_{10,4}$.

To analyze the algorithm, when a vertex is colored we expose all pairs involved in that vertex. In this way, the numbers Z_{rb} are always determined. Furthermore, W points are available for the pairs that will be exposed during the next step. At any step, the number of points not yet involved in exposed pairs is

$$W = 4Z_{00} + 3Z_{10} + 3Z_{01} + 2Z_{02} + 2Z_{20} + 2Z_{11} + Z_{12} + Z_{21} + Z_{03} + Z_{30}.$$

The goal is to compute the expected change $\Delta(Z_{rb})$ for any two consecutive steps, i.e. when exposing a new pair of points, each point in a different vertex.

When a vertex u is newly colored R and one of the pairs containing a point p in that vertex is exposed, the other point will be in some vertex v. The probability that v is of type (i, j) is $(4 - i - j)Z_{ij}/(W - 1)$.

Define d_{rb} to be the expected contribution to the increment $\Delta(Z_{rb})$ in Z_{rb} due to exposing a pair of points (i.e by coloring a vertex v either R or B). Up to terms $O(1/W)$, the gain of d_{rb} is $(4 - i - j)Z_{ij}/W$ when $(i, j) = (r - 1, b)$, and $(4 - (i + j))Z_{ij}/(W - 1)$ if $(i, j) = (r, b)$.

For instance, consider the vertex u in Figure 16 and let us see the effect on d_{20} of coloring vertex v with R. If vertex u is of type $(1, 0)$ (left figure), then u becomes (2,0), so Z_{20} increases by 1. Notice that the number of ways u can be $(1, 0)$ is 3. On the other hand, if u is $(2, 0)$ (right figure), then, when v gets R, u becomes $(3, 0)$ and Z_{20} decreases by 1. The number of ways u can be $(2, 0)$ is 2. Therefore,

$$d_{20} = \frac{3Z_{10} - 2Z_{20}}{W}.$$

We can extend the previous argument to get all the expected values d_{rb}, when a

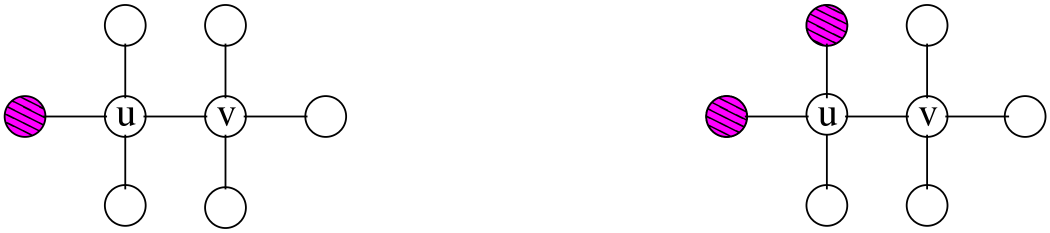

Figure 16: Effect on d_{20} of coloring vertex v with R

vertex v is colored R.

$$\begin{array}{lllll}
d_{00} = -\dfrac{4Z_{00}}{W} & d_{01} = -\dfrac{3Z_{01}}{W} & d_{02} = -\dfrac{2Z_{02}}{W} & d_{03} = -\dfrac{Z_{03}}{W} & d_{04} = 0 \\
d_{10} = \dfrac{4Z_{00} - 3Z_{10}}{W} & d_{11} = \dfrac{3Z_{01} - 2Z_{11}}{W} & d_{12} = \dfrac{2Z_{02} - Z_{12}}{W} & d_{13} = \dfrac{Z_{03}}{W} & \\
d_{20} = \dfrac{3Z_{10} - 2Z_{20}}{W} & d_{21} = \dfrac{2Z_{11} - Z_{21}}{W} & d_{22} = \dfrac{Z_{12}}{W} & & \\
d_{30} = \dfrac{2Z_{20} - Z_{30}}{W} & d_{31} = \dfrac{Z_{21}}{W} & d_{40} = \dfrac{Z_{30}}{W}. & &
\end{array}$$

The corresponding equations when a vertex is colored B form a symmetric set with these: they are the same but with the index pair on all variables swapped. Therefore, denoting by $\bar{d}_{r,b}$ the expected increments due to a dual step, consisting of coloring two vertices R and B, we get (ignoring $O(1/W)$ terms)

$$\begin{array}{lll}
\bar{d}_{00} = -\dfrac{8Z_{00}}{W} & \bar{d}_{01} = \dfrac{-6Z_{01} + 4Z_{00}}{W} & \bar{d}_{11} = \dfrac{3Z_{01} + 3Z_{10} - 4Z_{11}}{W} \\
\bar{d}_{02} = \dfrac{-4Z_{02} + 3Z_{01}}{W} & \bar{d}_{12} = \dfrac{2Z_{02} - 2Z_{12} + 2Z_{11}}{W} & \bar{d}_{22} = \dfrac{Z_{12} + Z_{21}}{W} \\
\bar{d}_{03} = \dfrac{-2Z_{03} + 2Z_{02}}{W} & \bar{d}_{13} = \dfrac{Z_{03} + Z_{12}}{W} & \bar{d}_{04} = \dfrac{Z_{03}}{W}.
\end{array} \tag{13}$$

In the same way, we can get symmetric equations for $\bar{d}_{rb}$ in the case that $r > b$.

At any given step in the algorithm, let ϕ_1 be the probability of processing a pair $(0,1)$-$(1,0)$, let ϕ_2 be the probability of processing a pair $(0,2)$-$(2,0)$, and let ϕ_3 be the probability of processing a $(2,1)$-$(1,2)$ pair. Then

$$\phi_1 + \phi_2 + \phi_3 = 1. \tag{14}$$

Moreover, every dual coloring of a $(0,1)$-symmetric pair produces in expectation $3\bar{d}_{02}$ $(0,2)$-symmetric pairs and $3\bar{d}_{12}$ $(1,2)$-symmetric pairs. Every dual coloring of a $(0,2)$-symmetric pair produces $2\bar{d}_{02}$ $(0,2)$-symmetric pairs and $2\bar{d}_{12}$ $(1,2)$-symmetric pairs; and every dual coloring of $(1,2)$-symmetric pair produces $\bar{d}_{02}$ $(0,2)$-symmetric pairs and $\bar{d}_{12}$ $(1,2)$-symmetric pairs. When coloring a pair $(2,1)$-$(1,2)$ the expected number of $(0,2)$-symmetric pairs produced in a given step is

$$(3\phi_1 + 2\phi_2 + \phi_3)\bar{d}_{02}, \tag{15}$$

In the same way,

$$(3\phi_1 + 2\phi_2 + \phi_3)\bar{d}_{12}. \tag{16}$$

Solving the system formed by (14), 15 and 16, we get

$$\phi_1 = \frac{1 - 2\bar{d}_{02} - \bar{d}_{12}}{1 + \bar{d}_{02} + 2\bar{d}_{12}}; \quad \phi_2 = \frac{3\bar{d}_{02}}{1 + \bar{d}_{02} + 2\bar{d}_{12}}; \quad \phi_3 = \frac{3\bar{d}_{12}}{1 + \bar{d}_{02} + 2\bar{d}_{12}}. \tag{17}$$

Notice phase 1 in Algorithm 4 will finish when $Z_{00} = Z_{02} = Z_{12}$, i.e. when $\phi_1 = 0$, which implies $\phi_2 + \phi_3 = 1$. Let us compute the expected increments of the random variables Z_{ij} at each iteration in phase 1,

$$\mathbf{E}\left(\Delta(Z_{ij})\right) = \bar{d}_{ij}(\phi_3 + 2\phi_2 + 3\phi_1) - \delta_{01}\phi_1 - \delta_{02}\phi_2 - \delta_{12}\phi_3,$$

where $i \leq j$, $i + j \leq 3$ and $\delta_{pq} = 1$ if $(p, q) = (i, j)$, and 0 otherwise.

Convert the previous difference equation in a differential equation by scaling by n and making $z_{ij} = Z_{ij}/n$, $x = t/n$, $W/n = w(x)$ and $A(x) = \theta_3 + 2\theta_2(x) + 3\theta_1(x)$ ($\theta_i(x)$ represents $\phi(t/n)$ and it is defined as ϕ_i in equation (17)),

$$\frac{dz_{00}}{dx} = -8z_{00}\frac{A(x)}{w(x)}, \frac{dz_{01}}{dx} = (4z_{00} - 6z_{01})\frac{A(x)}{w(x)} - \theta_1(x), \frac{dz_{11}}{dx} = (6z_{01} - 4z_{11})\frac{A(x)}{w(x)}, \quad (18)$$

where $w(x) = 4z_{00} + 6z_{01} + 4z_{02} + 2z_{11} + 2z_{12} + 2z_{03}$.

As it is proved in [25], $\theta_1(x) > 0$ through the whole phase 1, therefore the number of vertices of types $(0, 2)$, $(2, 0)$, $(1, 2)$ and $(2, 1)$ remain small and very seldom will appear symmetric pairs of these types, so very few vertices of types $(1, 3)$ or $(2, 2)$ are ever created, which implies that

$$\frac{dz_{ij}}{dx} = 0, \quad j \geq 2. \quad (19)$$

Moreover, as only the coloring of $(1, 2)$ or $(2, 1)$ contribute to the bisection in phase 1 (see Figure 17), the size of the bisection is approximately equal to the twice the total number of $(1, 2)$-symmetric processed pairs. Let $Y(t)$ be a random variable keeping track of the number of times a $(1, 2)$-symmetric pair is processed, the expected change in Y in one step is equal to $\bar{d}_{12}$. As before scale by n and let $y(x) = Y(t)/n$, then

$$\frac{dy(x)}{dx} = z_{11}\frac{A(x)}{w(x)}. \quad (20)$$

Solving the ODE consisting of (18), (4.3) and (20), with initial conditions

$$z_{00}(0) = 1; z_{ij}(0) = 0 \text{ for } 0 < i \leq j, i + j \leq 4; y(0) = 0,$$

we get a bisection of size asymptotically $2z_{11}n = n/3$.

Therefore, the solution of the differential equation system at the end of phase 1 represents the situation that all nodes are colored, and indicates a bisection of size asymptotically $2z_{11}n = n/3$. Algorithm 4 makes decisions according to certain *priorities* on the color of the neighbors. Then, using a non-prioritized algorithm, it can be formally shown that after the second phase, the bisection width $\mathfrak{B}$ is bounded below by $n/3$, which is the sharper existing lower bound on the bisection width of a random 4-regular graph. In [25], there is an easy argument to find and upper-bound for the bisection width of a random 4-regular graph.

The final result for random 4-regular graph is that a.a.s. the bisection width $\mathfrak{B}$ is bounded by $0.333333n \leq \mathfrak{B} \leq 1.66667n$. The algorithm for the 3-regular graph

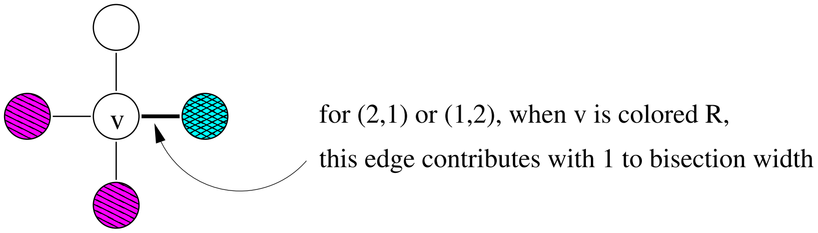
v
for (2,1) or (1,2), when v is colored R,
this edge contributes with 1 to bisection width

Figure 17: Contribution of $(2, 1)$ and $(1, 2)$ to the bisection width $\mathfrak{B}$.

is more involved and the result for the bisection width of 3-regular graphs is that a.a.s. $0.174039n \leq \mathfrak{B} \leq 1.32697n$ (see the paper for further details). It remains as an open problem to further close the gap for both, 3 and 4 random regular graphs. In a posterior publication [24], the method was extended to find bounds for the bisection width of any random r-regular graph. The paper also provides numerical bounds for values $5 \leq r \leq 12$.

4.4 Other work on $\mathcal{G}_{n,r}$, using the DEM

Given a graph $G = (V, E)$, a *dominating set* is a subset $D \subset V$ such that for any $v \in V$, either $v \in D$ or there is $u \in D$ such that $\{u, v\} \in E$. An *independent dominating set* is a dominating set D such that no two vertices of it are connected by an edge of E. Given a graph G, the *independent dominated set* problem is to find an independent dominating set with minimum size. The problem is NP-hard [42], even for 3-regular graphs. Duckworth and Wormald [34], using a random greedy algorithm proved that $0.2641n \leq |D| \leq 0.2794n$ a.a.s. for random 3-regular graphs. Later, the same authors extended their results to general $\mathcal{G}_{n,r}$ [33].

Given a graph $G = (V, E)$, an *induced matching* of G is a set of vertex disjoint edges $M \subseteq E$, such that no two edges in M have their end points connected by an edge. The problem of the *maximum induced matching* is given a graph G, find a maximum induced matching. The problem is NP-hard [74]. Duckworth, Wormald and Zito [32], gave a randomized greedy algorithm and using the DEM they proved that a.a.s. for a random 3-regular graph, $0.270413n \leq |M| \leq 0.282069n$.

5 Random k-SAT

A k-SAT-formula on n variables $\{x_1, \dots, x_n\}$ is a Boolean formula with n variables, which is a conjunction of m clauses, each of them being a disjunction of k *literals*, where a literal is a variable together with its positive or negative sign. For example, the following is a 3-SAT-formula with $m = 5$ and $n = 4$: $\phi = (\bar{x_1} \vee \bar{x_2} \vee \bar{x_4}) \wedge (\bar{x_1} \vee x_2 \vee x_3) \wedge (x_1 \vee \bar{x_2} \vee x_3) \wedge (x_2 \vee \bar{x_3} \vee x_4) \wedge (\bar{x_2} \vee x_3 \vee \bar{x_4})$.

The ratio $c = m/n$ is usually referred to as the *density* of a k-SAT-formula. A k-SAT-formula is *satisfiable* if there exists an assignment $\alpha \in \{0, 1\}^n$ to the n variables such that in each clause there exists at least one variable appearing positively and assigned a value of 1 under α or at least one variable appearing negatively and assigned a value of 0 under α. Notice that for a given n, the more clauses m a formula has, the more difficult is to satisfy the formula. Friedgut [38] proved that there is a sequence of reals $(c_n^k)_n$ such that for any $\epsilon > 0$ the probability of a randomly chosen k-SAT-formula on n variables with density $c_n^k - \epsilon$ being satisfiable approaches 1, as $n \to \infty$, whereas for density $c_n^k - \epsilon$, it approaches 0. Intuitively this means that formulae with many clauses are very likely to be unsatisfiable, whereas formulae with few clauses are very likely to be satisfiable, and that the transition from satisfiability to unsatisfiability is sharp. However, for values $k \leq 7$, it is still not known if $\{c_n^k\}_n$ converges. It is widely believed, though, see for example [21], that for any $k \geq 3$, there is a critical value c_k, such that, as $n \to \infty$, a k-SAT-formula with clause density $c_k - \epsilon$ is satisfiable a.a.s. and a k-SAT-formula with clause density $c_k + \epsilon$ is unsatisfiable a.a.s. For $k = 2$ it was proved by [21, 43, 36] that $c = 1$ is a sharp threshold. In parallel, research done using techniques from statistical physics (*Replica Symmetry Breaking, Cavity Method*), gave theoretical non-rigorous evidence for the thresholds of k-SAT. In particular, Mezard, Parisi and Zecchina [62, 63] stated that the threshold for 3-SAT occurs at clause density $c_3 = 4.27$. Recently, Achlioptas and Ricci-Tersenghi [5] proved that for $k \geq 8$, the thresholds for K-SAT found by use of statistical physics techniques could be considered rigorously proved. In the following we focus on methods giving increasingly improved rigorous lower and upper bounds on that threshold point.

The model used here is the *configuration model* with a *typical Poisson degree sequence*, where the formula is given by a sequence $(d_{i,j})_{i,j \in \mathbb{N}}$ of degrees, each $d_{i,j}$ representing the number of variables appearing i times positively and j times negatively. Each variable with i positive and j negative occurrences thus has $i + j$ distinct labeled copies, with i copies labeled positively and j copies labeled negatively. Figure 18 shows an example of a formula with degree sequence $d_{12} = 2, d_{23} = 1, d_{31} = 1$ on the variable set $X = \{x_1, x_2, x_3, x_4\}$. It is known, that for establishing lower and upper bounds, the configuration model with a typical Poisson degree sequence can be transferred to the more "classical models" of boolean formulae, see for example [31]):

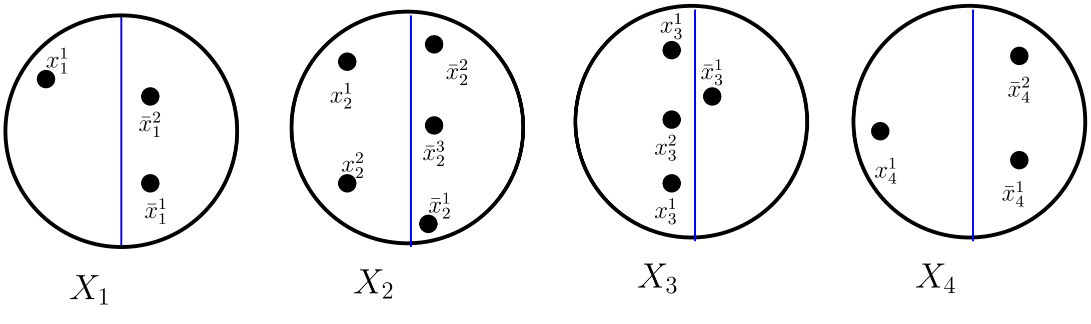

Figure 18: Example of labeled copies of 4 variables

A random k-SAT-formula is then chosen by choosing u.a.r. a partition of the set of

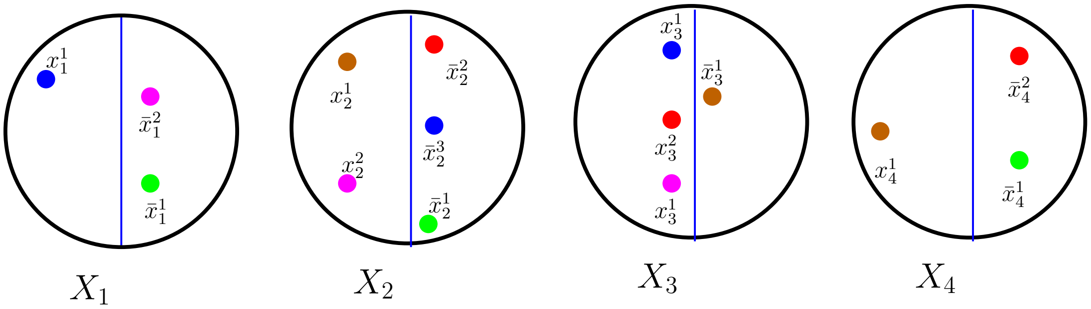

Figure 19: Example of a matching of the labeled copies of 4 variables

copies into sets of size k. By throwing away the labels of the copies, these sets of size k then become clauses. Figure 19 shows a possible partition of these copies producing the formula $\phi = (\bar{x_1} \vee \bar{x_2} \vee \bar{x_4}) \wedge (\bar{x_1} \vee x_2 \vee x_3) \wedge (x_1 \vee \bar{x_2} \vee x_3) \wedge (x_2 \vee \bar{x_3} \vee x_4) \wedge (\bar{x_2} \vee x_3 \vee \bar{x_4})$. Define a *multi-formula* as a SAT formula with repeated clauses and/or repeated variables per clause. Note that by the previous generation of SAT formulae multi-formulae might appear, but there is a non-zero probability of obtaining a *simple* formula, that is, a formula without repeated clauses or repeated variables in a clause. It is well known, that conditioning under having a simple formula, all formulae with the degree sequence given by the $(d_{i,j})_{i,j \in \mathbb{N}}$ are equally likely to be chosen under this model. Hence an algorithm rejecting all multi-formulae and choosing the first simple formula to be found, produces a formula u.a.r. among all formulae with the given degree sequence.

5.1 Lower bounds on random k-SAT using DEM

In this subsection we survey those lower bounds which were obtained by algorithms whose analysis is based on the method of differential equations.

5.1.1 Pure literal elimination rule

A *pure literal* is a literal whose complement does not appear in the formula. The *pure literal rule* for satisfying a k-SAT-formula is given in Algorithm 5.

Algorithm 5 The pure literal elimination rule

```
given $\phi = \{c_1, \ldots, c_m\}$ and variables $\{x_1, \ldots, x_n\}$
while there is a pure literal available
  choose u.a.r. one of them
  assign the variable corresponding to it so that all occurrences of that literal are satisfied
  remove all clauses containing that literal
if clauses are left then fail
```

This rule is a very conservative strategy, in that it only assigns a value to a variable that maintains the satisfiability of a formula.

Broder, Frieze and Upfal [18] proved that the pure literal rule a.a.s. succeeds in satisfying random 3-SAT-formulae with density up to 1.63, and they also proved that this rule fails for random 3-SAT-formulae with density greater than 1.7. Their original proof did not use the method of differential equations, but later a different proof using DEM was found by Mitzenmacher [64]. We will here survey here the approach of Mitzenmacher. Using the DEM, he found the exact density value for which this rule works. Moreover, his results easily extend to random k-SAT-formulae for any fixed value $k > 3$.

To analyze the pure literal rule using DEM, we assume that at every moment a pure literal is chosen u.a.r. from all pure literals available, and then the literal and also its negation are removed from the set of literals under consideration. In the analysis, we use the following variables (scaled by n) at time t, for $t = 0, \ldots, 1$: $\ell(t)$ denotes the scaled number of undeleted literals remaining, $x_i(t)$ is the scaled number of undeleted literals appearing i times in the formula, $c(t)$ is the scaled number of clauses remaining at time t, and $a(t)$ is the average number of clauses in which a literal chosen u.a.r. from all literals appears. Note that $x_0(t)$ is the scaled number of pure literals at time t (recall that a variable appearing zero times in the variables counts twice). If for some value of t we have $x_0(t) = 0$ but $c(t) > 0$, then the pure literal rule fails, as the pure literals have run out while clauses still remain. If, however, $x_0(t)$ goes to 0 as $c(t)$ goes to 0, then a.a.s. the pure literal rule succeeds. Since we are interested in the case when $n \to \infty$, each step of size dt can be thought of as one pure literal elimination (of size $1/n$ in the scaled variables). Thus the differential equations are only defined for those values of t where $x_0(t) > 0$; when $x_0(t) = 0$ the system stops.

The initial conditions are $\ell(0) = 2$, $c(0) = m/n$ and $x_i(0) = e^{-\mu}\mu^i/i!$ with $\mu = \frac{mk}{n}$, since for $n \to \infty$ the typical distribution of $x_i(0)$ is a Poisson distribution with parameter μ. As at each step two literals are removed, the pure literal and its negation, we have $\frac{d\ell}{dt} = -2$. By taking the initial condition $\ell(0) = 2$ into account, we obtain $\ell(t) = 2 - 2t$. When a pure literal is chosen u.a.r., the expected number of times it appears is $a(t)$ (up to a factor of size $O(1/n)$, a pure literal is equally likely to be any of the remaining variables). Thus,

$$\frac{dc}{dt} = -a(t) = \frac{-\sum_{i \geq 0} i x_i}{\ell} = \frac{-ck}{2-2t},$$

from which one obtains $c(t) = c(0)(1-t)^{k/2}$.

In a similar way, to obtain the differential equations describing the behavior of the $x_i(t)$, first observe that with probability x_i/ℓ, a pure literal deleted at some point appears i times. If the pure literal chosen appears j times, then apart from that literal all other $k-1$ literals of all j clauses are eliminated. The probability for each of those literals to appear i times is $\frac{ix_i}{ck}$ (up to $O(1/n)$ terms). Note that the number of literals appearing i times may also increase if one appearance of a literal appearing

$i+1$ times is deleted. Therefore, if $i \geq 1$ we obtain

$$\frac{dx_i}{dt} = -\frac{a(k-1)ix_i}{ck} + \frac{a(k-1)(i+1)x_{i+1}}{ck} - \frac{x_i}{\ell},$$

whereas for x_0, we have

$$\frac{dx_0}{dt} = \frac{a(k-1)x_1}{ck} - \frac{x_0}{\ell} - 1.$$

It can be checked that the solution is given by

$$x_i(t) = \sum_{j=1}^{\infty} \lambda_{i,j}(\frac{c(t)n}{m})^{j(k-1)/k}(1-t)^{1/2},$$

where $\lambda_{i,j} = \frac{2(\frac{mk}{2n})^j(-1)^(i+j)\binom{j}{i}}{j!}$, for $i \geq 1$. For x_0 we obtain

$$x_0(t) = 2(1-t)^{1/2}\left((1-t)^{1/2} + exp(\frac{-(1-t)^{(k-1)/2}ck}{2}) - 1\right),$$

with c being m/n. To determine when $x_0(t) > 0$, it suffices to check

$$(1-t)^{1/2} + e^{\frac{-(1-t)^{(k-1)/2}ck}{2}} - 1$$

and to find the largest c such that this expression is positive for all $t \in [0,1)$. Thus, we can can find the values of c and t for which the above expression and also its derivatives are 0 at t, and we get the numerical thresholds $c_3 = 1.636938..$, $c_4 = 1.544559..$, $c_5 = 1.403560..$, and so on. One can check that for any t not too close to the end, the hypotheses of Wormald's theorem are satisfied, and therefore the values of $x_i(t)$ and $c(t)$ are concentrated around their expected values.

5.1.2 Unit clause rule

The unit clause rule was first proposed and analyzed by Chao and Franco [20]. The basic idea is explained in Algorithm 6.

Chao and Franco use the DEM as the main tool to analyze their algorithm. In particular, at every step they keep track of the number of 2-clauses and the number of 3-clauses generated in this step. If at some point the density of the underlying 2-SAT-formula is above 1, then the formula is not satisfiable anymore. It turns out that this condition is also asymptotically necessary for a random 3-SAT-formula to be unsatisfiable. The intuitive explanation is the following: if the density of 2-clauses in each step is below 1, unit clauses do not accumulate, and thus the probability that the algorithm fails in one step by producing a 0-clause is small.

Let $t_e = (1-\epsilon)n$. The authors show that if at each of the $t \leq t_e$ iterations of the algorithm, the expected number of 2-clauses is below the number of variables remaining at that step, then with positive probability during these steps the algorithm

Algorithm 6 The unit clause rule

```
given $\phi = \{c_1, \ldots, c_m\}$ and variables $\{x_1, \ldots, x_n\}$
while there are some clauses left
  if there is an empty clause then fail
  else if there is a unit clause remaining
    choose u.a.r. one of them, and satisfy it
  else
    pick an unset variable u.a.r.
    assign it to true/false u.a.r.
    remove all clauses satisfied by this variable
    shrink all clauses where this variable appears with the opposite sign
```

does not fail, i.e., no 0-clause is generated during these steps. Moreover, at the t_e'th step no 0-clause and no 1-clause is left. Regarding the evolution of 3-clauses and 2-clauses, denote by $C_i(t)$ the number of i-clauses at time t. Observe that one particular 3-clause is eliminated at each step t with probability $3/(n-t)$, as there are $n-t$ unset variables remaining at step t and 3 of them appear in the 3-clause. Thus, the expected number of 3-clauses decreases at step t by $3C_3(t)/(n-t)$. If no unit clauses exist at step t, this is clear from the definition of the algorithm; if a unit clause exists, it can be shown that the literal chosen to satisfy a unit clause chosen u.a.r. is uniformly distributed among all literals. Similarly, the expected change of the number of 2-clauses at step t is $\frac{-2C_2(t)}{n-t} + \frac{3C_3(t)}{2(n-t)}$, where the second term comes from the fact that a 3-clause deleted becomes a 2-clause with probability $\frac{1}{2}$ (recall that the variable chosen is assigned its sign u.a.r.). Scaling variables as before ($x = t/n$), and using lowercase letters for the scaled versions of the variables, we get the following differential equations:

$$\frac{dc_3}{dx} = -\frac{-3c_3(x)}{1-x}, \quad \frac{dc_2}{dx} = \frac{-2c_2(x)}{1-x} + \frac{3c_3(x)}{2(1-x)},$$

with initial conditions $c_2(0) = 0$ and $c_3(0) = c = m/n$. Solving these equations one obtains $c_3(x) = (1-x)^3$ and $c_2(x) = \frac{3}{2}cx(1-x)^2$, and by Wormald's theorem we obtain that for any $0 \leq t \leq t_e$ and for $i = 2, 3$, $C_i(t) = nc_i(t/n) + o(n)$ a.a.s. If $c \geq \frac{8}{3}(1+\delta)$, then $c_2(\lfloor n/2 \rfloor) \geq (1 - \frac{\delta}{2})n/2$ for any $\delta > 0$, and thus a.a.s. the algorithm fails to produce a satisfying assignment. If $c \leq \frac{8}{3}(1-\delta)$, at every point $t \leq t_e := \frac{9}{10}n$, $C_2(t) \leq (1-\delta)(n-t)$, and by the above observations, with positive probability the algorithm does not fail until t_e. From then on, just remove a randomly chosen literal from each 3-clause to obtain a 2-SAT-formula with $\frac{3}{4}(n-t_e)$ clauses, which is satisfiable a.a.s.

The same analysis can also be applied for general k-SAT-formula. In this case the algorithm produces a satisfying assignment for $c_k \leq \alpha 2^k/k$, where $\alpha = \frac{1}{2}(\frac{k-1}{k-2})^{k-2}$ (see [2]). Moreover, Chao and Franco in their paper presented the following improvement of the unit clause rule, given in Algorithm 7. The idea is that the number of

Algorithm 7 The improved unit clause rule

```
given $\phi = \{c_1, \dots, c_m\}$ and variables $\{x_1, \dots, x_n\}$
while there are some clauses left
  if there is an empty clause then fail
  else if there is a unit clause remaining
    choose u.a.r. one of them, and satisfy it
  else
    pick an unset variable u.a.r.
    if this variable appears positively in at least half of the 3-clauses set it to 1
    else set it to 0
    remove all clauses satisfied by this variable
    shrink all clauses where this variable appears with the opposite sign
```

3-clauses that become 2-clauses is now smaller, and thus the algorithm succeeds even for higher values of c. The analysis is more difficult now, as the expected change in the number of 2-clauses depends on the existence of unit clauses at that step. One way to get rid of this dependence is given in Algorithm 8, where at each step a coin is tossed to decide whether or not the unit clause rule should be applied.

Algorithm 8 The modified improved unit clause rule

```
given $\phi = \{c_1, \dots, c_m\}$ and variables $\{x_1, \dots, x_n\}$
while there are some clauses left
  toss a coin $C$ which is 1 with probability $u = \min\{(1+\delta)C_2(t)/(n-t), 1\}$
  if there is an empty clause then fail
  else if $C = 1$ and there is a unit clause remaining
    choose u.a.r. one of them, and satisfy it
  else if coin=1 and there is no unit clause
    choose a variable u.a.r. and set it a value u.a.r.
    remove all clauses satisfied by this variable
    shrink all clauses where this variable appears with the opposite sign
  else
    pick an unset variable u.a.r.
    if this variable appears positively in at least half of the 3-clauses then set it to 1
    else set it to 0
    remove all clauses satisfied by this variable
    shrink all clauses where this variable appears with the opposite sign
```

In this way the differential equations get easier. in fact, a further simplification helps a lot: in each of the steps that happen with probability $1 - u$, the algorithm can toss another coin to decide whether to apply the majority rule or whether selecting the sign of the variable chosen to be u.a.r. By setting up similar equations, one obtains that this algorithm works for random 3-SAT-formulae up to density 2.99 ([20]) or, using a slightly improved analysis, even up to density 3.001([2]).

5.1.3 Further algorithmic improvements using DEM

All further improvements on obtaining lower bounds for 3-SAT-formulae stem from the analysis of refined algorithms using differential equations. First, Frieze and Suen [41] analyzed algorithms exploiting clause length. Their algorithm picks always a shortest clause u.a.r. from all shortest clauses, chooses a literal from that clause u.a.r. and satisfies it. Using similar ideas as in the previous subsections, they show that their algorithm satisfies formulae with density up to 3.003. For technical reasons in the analysis of Wormald's theorem it turns out to be helpful to start with additional δn 2-clauses, since otherwise at some early step the algorithm might "run out" of 2-clauses. Later, Achlioptas [1] improved this bound to density 3.14 by combining majority rule with clause length. Achlioptas and Sorkin [6] further improved this bound to $c = 3.26$. Finally, Kaporis, Kirousis and Lalas [50] and independently Hajiaghayi and Sorkin [46], improved these results up to $c = 3.52$ by taking into account literal degrees. Hajiaghayi and Sorkin choose those literals first which maximize the absolute value of the difference of positive and negative occurrences of the literals, with ties being broken in favor of literals appearing in total more often. Kaporis, Kirousis and Lalas choose the literal in a step without unit clauses slightly differently. Intuitively speaking, they first choose the bucket of the literals from which the literal is chosen, where the (i, j)-th bucket contains all literals with i positive and j negative occurrences (indeed literals, not variables; that is, if a literal γ belongs to bucket $B_{i,j}$, then the literal $\bar{\gamma} \in B_{j,i}$, and thus clearly $|B_{i,j}| = |B_{j,i}|$). Out of the literals in one bucket one literal is chosen u.a.r. (say). The (i, j)-th bucket is chosen if i and j are those values such that in all the steps without unit clauses following the current step relatively many 3-clauses remain, which might become possible 2-clauses, and relatively few 2-clauses emerge, which might become 1-clauses.

5.2 Usage of DEM for obtaining upper bounds of random 3-SAT

The method of DEM has also been used, in combination with other methods, to improve upper bounds of random 3-SAT. The known upper bounds for random 3-SAT all use the so called *first moment method*: the probability of the existence of a satisfying assignment of a given random 3-SAT-formula is bounded from above by the expected number of satisfying assignments of that formula. A naive application of this method gives an upper bound of 5.19. Further improvements can be made by not counting all satisfying assignments and by restricting the input space of formulae to typical formulae (see for example [52, 31]). The authors of [31] obtain a bound of 4.506 by counting only satisfying assignments which are maximal w.r.t. a certain ordering of the assignments and counting only formulae whose degree sequence of variables is typical, that is, for every i and j, the number of variables appearing i times positively and j times negatively is close to the expected number[1]. In [27] the

[1] In fact, the authors of [31] also perform a certain "positive unbalancing" operation of the variables, that is, they flip all variables in such a way that the number of positive occurrences is at least the number of negative occurrences. It turns out that in this way the number of satisfying assignments maximal w.r.t. the ordering considered there decreases.

authors improve the bound of Dubois to 4.4898 using DEM. The bound is rigorous up to one 3×3-system of equations which is solved only numerically. The idea of this paper is the following: instead of counting special satisfying assignments of the original formula, at first, for the purpose of the analysis, the pure literal rule (see Algorithm 5) is applied to the formula. To obtain the new typical degree sequence of this *impure core* of the original formula, the authors make use of the DEM. Denote by $Y_{i,j}(t)$ the number of variables with i positive and j negative occurrences at time t, by $Y_0(t) = \sum_{i>0} iY_{i,0}(t) + \sum_{j>0} jY_{0,j}(t)$ the total number of pure literal occurrences at time t, and denote by $L(t)$ the total number of remaining literals at time t. As before, define the corresponding scaled versions $y_{i,j}(t) = \frac{1}{n}Y_{i,j}(tn)$, $y_0(t) = \frac{1}{n}Y_0(tn)$ and $\ell(t) = \frac{1}{n}L(tn)$. Thus, when eliminating a randomly chosen pure literal, we obtain (similarly to the equations given in Section 5.1.1) the following differential equations:

$$\begin{aligned}
\frac{d\ell}{dt} &= -3 \\
\frac{dy_{i,j}}{dt} &= \frac{2}{\ell}\left((i+1)y_{i+1,j} + (j+1)y_{i,j+1} - (i+j)y_{i,j}\right), \quad i,j > 0, \\
\frac{dy_0}{dt} &= \frac{2}{\ell}\left(\sum_{i>0} iy_{i,1} + \sum_{j>0} y_{1,j} - y_0\right) - 1,
\end{aligned}$$

with the initial conditions $\ell(0) = 3c$ (c being m/n), $y_0(0) = \sum_{i>0} id_{i,0} + \sum_{j>0} jd_{0,j}$, and for $i,j > 0$ we have $y_{i,j}(0) = d_{i,j}$, where $d_{i,j}$ denotes the typical scaled number of variables appearing i times positively and j times negatively in a random 3-SAT-formula. One can check the conditions of Wormald's theorem (see [27] for details) and obtains as unique solution $\ell(t) = 3c - 3t$,

$$y_{i,j}(t) = \sum_{k \geq i} \sum_{\ell \geq j} d(k,\ell) \binom{k}{i} \binom{\ell}{j} (b(t))^{i+j} (1 - b(t))^{k-i} (1 - b(t))^{\ell - j},$$

where $b(t) = (1 - \frac{t}{c})^{2/3}$. Also, $y_0(t) = 3c - 3t - \sum_{i,j>0}(i+j)y_{i,j}(t)$. The system is solved for the first point t such that $y_0(t/n) = 0$, which is the point when the last pure literal disappears. Applying Dubois' ideas on the new formula together with additional typicality constraints on the type of clauses the authors of [27] obtain the bound 4.4898, which is the best upper bound currently known. Maneva and Sinclair [57] proved a better upper bound of 4.453, however using a conjecture of the geometry of the solution space of satisfying assignments which is only known to be true for k-SAT, if $k \geq 8$.

6 Sudden emergence of the k-core in $G_{n,p}$

The k-core of a graph G is defined as the largest subgraph of G with the property of every vertex having degree at least k. It is easy to see that it is unique and that it can be found by repeatedly deleting all vertices of degree less than k. During these deletions, the degrees of other vertices might fall below k, the process continues, and so on, until no vertices of degree less than k remain. Note that the k-core can be empty. In Figure 20 we present an example of a graph together with its 3-core and its 4-core.

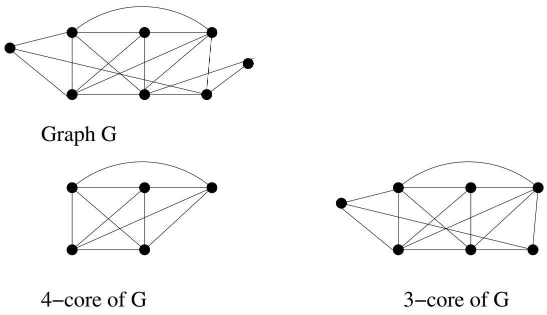

Figure 20: A graph together with its 3-core and its 4-core

We give an overview of the proof that for $G_{n,p}$ there exists some $c > 0$ such that for any fixed $k \geq 3$, the k-core is a.a.s. empty if $p < \frac{c}{n}$ and the core has at least a linear number of vertices if $p > \frac{c}{n}$. For the 2-core, things are different, easier and known for longer, see for example [37]. As usual, n is fixed, and we are interested in the behavior as $n \to \infty$. This result was first proven by [70], and we follow the exposition of this proof given in [77]. Later, additional results on the k-core, using different methods, were independently obtained by [19], by [48], and also by [71]. The authors of [77] choose a graph G u.a.r. from the $G_{n,m}$ model, that is, the model with m edges. It is well known, that for many properties such as the existence of a k-core, the results from $G_{n,m}$ can be easily transferred to $G_{n,p}$ (see for example [49]). We will also assume that $c_1 n \leq m \leq c_2 n$ for some constants $c_1, c_2 > 0$. For ease of the analysis we will assume that the algorithm for obtaining the k-core does not delete vertices, but rather only edges incident to a chosen vertex. Moreover, the vertex to be eliminated is chosen u.a.r. from all non-isolated vertices of degree $< k$, see Algorithm 9 for details. Now, define a vertex to be *light* if it is of degree strictly less than k, and *heavy* otherwise. Define the vector $\mathbf{w}(t) = (v_0(t), \ldots, v_{k-1}(t), m(t))$, where $v_i(t)$ denotes the number of vertices of degree i, $0 \leq i \leq k-1$, and $m(t) = |E(G_t)|$. Define furthermore by $v = \sum_{i=1}^{k-1} v_i$ ($\bar{v} = n - \sum_{i=0}^{k-1} v_i$, respectively) the number of non-

Algorithm 9 Algorithm for obtaining the k-core

```
given $G$ u.a.r. from $G_{n,m}$ and a fixed $k \geq 2$
**while** there exists some vertex of degree $1 \leq d \leq k-1$
  Let $L$ be the set of vertices of degree $1 \leq d \leq k-1$
  choose a vertex $v$ u.a.r. from $L$
  delete all edges incident to $v$
  update the degrees of other vertices
output the graph on the non-isolated vertices
```

isolated light vertices (the number of heavy vertices, respectively) and by $s = \sum_{i=1}^{k-1} iv_i$ ($\bar{s} = 2m - s$, respectively) the total degrees of light vertices (total degrees of heavy vertices, respectively). The process ends when $\bar{v} = n - v_0$, at which point there are no non-isolated light vertices remaining.

Observe that $\mathbf{w}(0)$ is a random vector and that conditional upon $\mathbf{w}(0)$, the distribution of G_0 is uniform in the set $G(\mathbf{w}(0))$, where $G(\mathbf{w})$ is the set of graphs with vector $\mathbf{w}$. By induction, given $\mathbf{w}(t)$, G_t is uniformly distributed in $G(\mathbf{w}(t))$, and hence the sequence $(\mathbf{w}(t))_t$ is a Markovian. The transition probabilities of $(\mathbf{w}(t))$ can be determined by counting the number of graphs with given degree sequence. This is rather complicated (see [77] for details), and we will just give the resulting scaled differential equations: writing $\nu_i(x) = v_i(xn)/n$, $\nu(x) = v(xn)/n$, $\mu(x) = m(xn)/n$, $\bar{\nu} = \bar{v}(xn)/n$ and $\sigma(x) = s(xn)/n$, one obtains the following system of differential equations:

$$\begin{aligned}
\frac{d\nu_i}{dx} &= \delta_{i0} + \frac{(i+1)\nu_{i+1}\sigma}{2\mu\nu} - \frac{i\mu_i\sigma}{2\mu\nu} - \frac{\nu_i}{\nu}, \quad 0 \le i < k-1 \\
\frac{d\nu_i}{dx} &= \frac{z^k \bar{\nu}\sigma}{2\mu\nu(k-1)! e_k(z)} - \frac{i\nu_i\sigma}{2\mu\nu} - \frac{\nu_i}{\nu}, \quad i = k-1 \\
\frac{d\mu}{dx} &= -\frac{\sigma}{\nu},
\end{aligned}$$

where $e_k(z) = \sum_{i \ge k} z^i/i!$ and z is such that $\bar{\nu}\mathbf{E}\left(X(z)\right) = 2\mu - \sigma$, and $X(z)$ being distributed as a Poisson variable with parameter z conditioned on being at least k (in other words, for $i \ge k$, $\mathbf{Pr}\left[X(z) = i\right] = \frac{\mathbf{Pr}[Z(z)=i]}{\mathbf{Pr}[Z(z) \ge k]}$, where $Z(z) \sim \mathbf{Po}(z)$.) Pittel et al. [70] have not been able to solve all differential equations, but they can show that $\frac{z^2}{\mu} = C_1$ and $\frac{e^z \bar{\nu}}{e_k(z)} = C_2$ for some constants C_1, C_2 depending on the initial conditions, which is enough to extract the information for the k-core. Define the domain D of Wormald's theorem to be the set of $(\nu_0, \ldots, \nu_{k-1}, \mu, x)$ such that

$$\mu > \epsilon_1,\ \bar{\nu} > \epsilon_1,\ 2\mu - \sigma > (k + \epsilon_1)\bar{\nu},\ \nu > \epsilon_2,$$

for small constants ϵ_2, ϵ_1 (and ϵ_2 much smaller than ϵ_1). One can also restrict the input graphs to those graphs having a degree sequence very close to the typical one (a truncated Poisson) and with not too large maximum degree.

We first give some semi-formal reasoning to get an idea at which point a non-empty k-core emerges: consider a branching process which starts with a single particle x_0, where the number of each particle has a Poisson distribution with mean c, and these numbers are independent for different particles. Let $\mathcal{B}_d$ be the event that this process contains a $(k-1)$-ary tree of height d with x_0 as a root, and let $\mathcal{B} = \lim_{d \to \infty} \mathcal{B}_d$. Denote by $\beta(c)$ the corresponding probability. Similarly, let $\mathcal{B}_d^+$ be the event that x_0 has at least k children, each of which has at least $k-1$ children, each of which has at least $k-1$ children,..., up to height d, let $\mathcal{B}^+ = \lim_{d \to \infty} \mathcal{B}_d^+$, and denote by $\beta^+(c)$ the corresponding probability. Finally, let $\gamma_k := \inf\{c : \beta(c) > 0\}$. Letting $\Psi_{\ge t}(c) = \mathbf{Pr}\left[\mathbf{Po}(c) \ge t\right]$, one can observe that by independence, $\mathbf{Pr}\left[\mathcal{B}_{d+1}\right] = \Psi_{\ge k-1}(c\mathbf{Pr}\left[\mathcal{B}_d\right])$.

Hence we also have

$$\begin{aligned} \beta(c) &= \Psi_{\geq k-1}(c\beta(c)), \qquad (21) \\ \beta^{+}(c) &= \Psi_{\geq k}(c\beta(c)). \end{aligned}$$

The functions $\beta(c)$ and $\beta^{+}(c)$ have a jump at $c = \gamma_k$ and are continuous for $c \geq \gamma_k$. If $\beta(c) > 0$, then using the notation $\lambda = c\beta(c)$, the first equation of (21) can be written as

$$c = \frac{\lambda}{\Psi_{\geq k}(\lambda)}. \qquad (22)$$

Using this notation once again, it is easy to see that γ_k satisfies $\gamma_k = \inf\{\frac{\lambda}{\Psi_{\geq k}(\lambda)} : \lambda > 0\}$. For values of $c < \gamma_k$ clearly Equation (22) has no root for λ. It turns out that γ_k is the threshold of the emergence of the k-core. In fact, for $c \geq \gamma_k$ the size of the k-core is $n\beta^{+}(c) + o(n)$ a.a.s. Now we give again a semi-formal intuition to prove that γ_k is indeed the threshold. It was already proven by [56], that for $k \geq 3$, a k-core, if it is non-empty, always contains a fraction of $0.0002n$ vertices. This is in sharp contrast to $k = 2$, where the expected length of the first cycle, which is equivalent to the emergence of a non-empty 2-core, is $Kn^{1/6}$ for $K = 2.0337$, as proven by [37]. For $c < \gamma_k$, it can be shown that the solution cannot exit at the boundary where $2\mu - \sigma = (k + \epsilon_1)\bar{\nu}$ or at the boundary where $\nu = \epsilon_2$. Thus, it must exit with either $\mu(x) = \epsilon_1$ or $\bar{\nu}(x) = \epsilon_1$, and up to this point one has $\nu > \epsilon_2$. That is, at time x there are still more than $\epsilon_2 n + o(n)$ light vertices left, but at most $\epsilon_1 n$ heavy vertices. Choosing the constants appropriately, $\epsilon_1 < 0.0002$, and thus the k-core is empty. On the other hand, for $c > \gamma_k$, the differential equations leave the domain when $\nu(x) = \epsilon_2$. From this point on, the Lipschitz condition fails, but the expected change in σ is still negative, and by applying a standard martingale type argument with stopping times one can show that the process finishes in at most Cn steps for some C which can be made arbitrarily small (see [70]).

7 Almost all graphs with average degree ≤ 4.003 are 3-colorable

Achlioptas and Moore [3] showed using a modification of Brélaz's heuristic [17], that a graph with average degree $d \leq 4.003$ is a.a.s. 3-colorable. More precisely, given a graph G chosen u.a.r. from all graphs in $\mathbb{G}_{n,d/n}$, they analyze Algorithm 10, which is backtracking-free, using the DEM. Initially, each vertex v has a list of three colors $\ell(v) = \{R, G, B\}$ available. At each step, select randomly a vertex v and a random color for the vertex, among the list of colors of v. Delete the selected color from the list of colors of the neighbors of that vertex. If at some step for some vertex v we have $\ell(v) = \emptyset$, the algorithm fails. Denote by a q-color-vertex a vertex v with $|\ell(v)| = q$. Throughout the section we denote by $deg(v)$ the number of not yet colored neighbors of v. The function h in Algorithm 10 is used to give priority to vertices with many not yet colored neighbors to be colored first; in particular, Achlioptas and Moore choose $h(i) = i^{\rho}$ for some sufficient large ρ.

Algorithm 10 Algorithm for 3-coloring a random graph

```
given $G \in_{u.a.r.} \mathbb{G}_{n,d/n}$
while there are uncolored vertices
  if there are 2-color vertices
    then
      pick a 2-color vertex $v$ with probability proportional to $h(deg(v))$
      pick a color $c \in \ell(v)$ u.a.r. and for all $w \in N(v)$: $\ell(w) := \ell(w) \setminus \{c\}$
    else
      pick a 3-color vertex $v$ u.a.r.
      pick a color $c \in \ell(v)$ u.a.r. and for all $w \in N(v)$: $\ell(w) := \ell(w) \setminus \{c\}$
  while there are 1-color vertices
      pick a 1-color vertex $v$ u.a.r.
      color it with the only color $c$ remaining, for all $w \in N(v)$ set $\ell(w) := \ell(w) \setminus \{c\}$
```

In their proof, the authors rely on the fact that the degree sequence of a graph in $\mathbb{G}_{n,d/n}$ is close to an expected Poisson degree sequence with maximum degree $O(\log n / \log\log n)$. In fact, they prove their result in the configuration model explained in Section 4, with fixed degree sequence, i.e. when each vertex v of degree d_v has d_v initially unexposed copies (denoted *half-edges*) which are matched with other not yet exposed copies. Those copies might belong to the same vertex, and these matchings might generate multiple edges, but it can be shown that there is positive probability to obtain a simple graph. Moreover, conditioned on having a simple graph, all graphs with the given degree sequence are equally likely.

To apply the DEM to the analysis of the algorithm, the authors first deal with vertices of degree $d > \Delta = 30$ separately: Define by E_H the set of edges incident to vertices of degree $> \Delta$, and define by H the graph induced by these edges. Furthermore, denote by Y the set of vertices of degree $\leq \Delta$ lying in cyclic components of H, and by E_Y the set of edges incident to vertices in Y. Denote by K the graph induced by $E_H \cup E_Y$, and set B to be the graph induced by the remaining edges. Finally, denote by L the set of vertices of degree $\leq \Delta$ in K which are not in Y. Intuitively speaking, K is the *high-degree graph*, B is the *low-degree graph*, and L is the interface between them. The typical degree sequence of B can be computed from the original typical degree sequence. It can be shown that H contains a.a.s. no multicylic components, no components of size $(\log n)^2$, and no more than $\log n$ unicyclic components. Also, a.a.s. no edge in E_Y connects two vertices belonging to Y and a.a.s. no vertex in $K \setminus H$ is contained in more than one edge in E_Y. It is then possible to show that K can be 3-colored in such a way that all vertices in L have monochromatic neighborhoods. Moreover, $|L| \leq \delta n$, where $\delta = \delta(\Delta)$ is some sufficiently small constant. Hence, by uncoloring the vertices of L, one can assume that the graph remained to 3-color is such that all vertices have degree $\leq \Delta$, for a $(1-\delta)$-fraction of them 3 colors are available, and for the remaining δn vertices there are still 2 colors available.

For the analysis of Algorithm 10, we need more notation. Define one *round* (one time unit in the algorithm) as one execution of the inner *while*-loop, that is, a sequence of steps starting with a *free* step (a step corresponding to coloring a 2-color-vertex) followed by a sequence of *forced* steps, corresponding to coloring 1-color-vertices. Hence there are at most n rounds. During each round, there may be several *iterations*, each one corresponding to the coloring of one vertex. Denote by $C_i(t)$, $C \in \{R, G, B\}$, $0 \leq i \leq \Delta$, the number of vertices of degree i at time (round) t which are 2-color-vertices without color C. Denote by $U_i(t) = R_i(t) + G_i(t) + B_i(t)$ the total number of 2-color-vertices at time t, and by $W_i(t)$ the number of 3-color-vertices at time t. Denote by $E(t)$ the number of unexposed copies (half-edges) at time t, by $U(t) = \sum_i iU_i(t)$ the number of unexposed copies belonging to 2-color-vertices at time t, and let $H(t) = \sum_i h(i)U_i(t)$ the sum of preferences of all 2-color-vertices at time t. Also define a list sequence of n vertices to be (δ, ϵ)-easy if it has maximum degree $O(1)$, $\sum_{i \geq 0} i(i-2)(U_i(t) + W_i(t)) \leq \epsilon n$ (intuitively this means that there are many vertices of degree 1), if all lists have at least 2 colors, and for every degree $i \geq 2$, there are distinct color lists ℓ_1, ℓ_2 such that at least δn vertices of degree i have color list ℓ_1 and at least δn vertices of degree i have color list ℓ_2. It is well known that such graphs a.a.s. have no multicyclic components and at most $L = L(\Delta)$ many cycles. Also, with constant probability, every cycle receives at list two different color lists, and hence it is easy to 3-color vertices with such list sequences. Hence, in the remainder the goal is to show that for the graph $B \cup L$ the corresponding degree sequence is (δ, ϵ)-easy for some $\delta, \epsilon > 0$.

Given a typical degree sequence of B with all degrees bounded by Δ, the first goal is to bound the probability that a 0-color-vertex is generated. The idea is to show that in a sequence of z forced steps the probability to generate a 0-color vertex is proportional to z^2/n. To see this, observe that in a graph with maximum degree Δ with δn unexposed copies in one single iteration of a round the probability that the half-edge exposed in this iteration hits another half-edge of a 1-color-vertex is at most $\frac{z\Delta}{\delta n}$. If no such matching occurs, the algorithm never generates a 0-color-vertex. Thus, it is sufficient to show that at each round, as long as vertices of $L \cup B$ are colored, the number of unexposed copies is $\Theta(n)$ and that the expected value of z^2 is bounded by some constant. In such a case, Algorithm 10 fails with probability at most $O(1/n)$ in each round, and as there are at most n rounds, there is constant probability to succeed in all rounds.

The key idea in the analysis of the proof is the following, each round is considered as a branching process, whose progenitor is the vertex chosen in the free step initiating this round. For each vertex v, the *progeny* of v is the set of 1-color-vertices created by coloring v. Since vertices with different degrees and different colors assigned produce different progeny in expectation, the authors deal with a multitype branching process, where a vertex type corresponds to a pair of degree and assigned color, and the whole process can be represented by a matrix M which captures all this information as follows: $M_{i,j}$ is the expected progeny of type-i when the progenitor is a vertex of type j. Multitype branching process theory states that if the largest eigenvalue of M is at most $1 - \delta$ for some $\delta > 0$, then the branching process is subcritical and a.a.s. dies out. Thus, in such a case, the length of z decreases geometrically, and hence it is bounded by some constant. In this case the following three facts can be shown as well.

1. If for every type i the probability of having more than t children is at most $(1-\theta)^t$, for some $\theta > 0$, then there exists some $\sigma = \sigma(\theta) > 0$ such that the probability of having more than s children in total is at most $(1-\sigma)^s$.
2. Letting p being the vector whose i-th entry represents the probability that the progenitor is of type i, it is not hard to show that the vector $m(p)$ whose i-th entry is the expected total progeny of the i-th type can be expressed as $(I-M)^{-1}p$ (I being the identity matrix) if $\lambda(M) < 1$.
3. Furthermore, $m(p)$ is smooth: if a different matrix M' with probability distribution q, capturing the same information as M, satisfies $|M - M'|_2 \le \epsilon$, if all entries of M, M' are bounded, if both $\lambda_1(M)$ and $\lambda_1(M')$ are bounded away from 1, and if $|p-q|_2 \le \zeta$, then $|m(p) - m'(q)|_2 \le L(\epsilon + \zeta)$ for some constant L independent of ϵ and ζ.

The third issue is important in the analysis of Algorithm 10, as in each round the progenitor distributions vary slightly. When representing the steps of the algorithm as a branching process, another issue that has to be taken care is fact that the graph contains cycles, and hence this is not a typical branching process. It can be shown, however, that in each round at most $O(\log n)^2$ copies are exposed, hence these issues affect only lower order terms.

For the analysis of the algorithm, the matrix M is now defined as follows: $M_{(x,i),(y,j)}$ denotes the expected number of 1-color-vertices of color x and degree i generated by assigning color y to a vertex of degree j. Denoting by $C_{i+1}(t)$ the number of 2-color-vertices of degree $i+1$ with available colors $\{x, y\}$, observe that for $x \neq y$

$$M_{(x,i),(y,j)} = \frac{j(i+1)C_{i+1}(t)}{E(t)},$$

as at time t $E(t)$ copies are unexposed, and in order to generate a vertex of degree i with single color x that vertex before must have degree $i+1$ and it must have two colors x, y available. Thus, the total number of unexposed copies corresponding to such vertices is $(i+1)C_{i+1}(t)$, and as there are j unexposed copies of the generating vertex, the result follows. If $x = y$, then $M_{(x,i),(y,j)} = 0$, because assigning color y to v cannot give rise to a neighbor of v with only color y available. A list sequence of colors corresponding to n vertices at time t is said to be (α, β)-*subcritical* if there exist $\alpha, \beta > 0$ such that $U(t) > \alpha n$ and for the corresponding matrix M, $\lambda_1(M) \le 1 - \beta$. If it holds that for all time units of Algorithm 10 the process is (α, β)-subcritical, then with positive probability the algorithm does not fail in these time units, i.e., it generates no 0-color-vertices. One just has to observe that the probability of failing in one round is bounded by the ratio of the number of unexposed copies belonging to vertices that have had at least one copy exposed during the current round divided by the total number of unexposed copies belonging to all other vertices and apply standard probabilistic arguments.

In order to set up the differential equations, let us first assume that the list sequence is (α, β)-subcritical, hence implying the existence of 2-color-vertices. First define by $p_{(c,i)}$ the probability that a vertex to be colored in a free step of the algorithm (hence being a 2-color-vertex) receives color $c \in \{R, G, B\}$ and it has degree i. Assume $c = B$ (the other cases follow analogously). Observe that the probability to choose a 2-color-vertex of degree i is $\frac{h(i)U_i(t)}{\sum_i h(i)U_i(t))} = \frac{h_i U_i(t)}{H(t)}$, and the chosen vertex receives color B with probability $\frac{1}{2}$, if B is one of the two available colors of that vertex. The probability of the latter case is $\frac{R_i(t)+G_i(t)}{U_i(t)}$. Thus we get

$$p_{(B,i)} = \frac{1}{2}\frac{h(i)(R_i(t) + G_i(t))}{H(t)}.$$

Denote by p now the vector whose entries are these probabilities. The authors show the following lemma

Lemma 3. *Let* $\mathrm{L}(t)$ *be a* (α, β)*-subcritical list sequence.*

1. *The expected number of copies exposed while coloring vertices with color* B *is* $k_B + o(1)$*, where*
$$k_B = \sum_{i=0}^{\Delta} i((I - M)^{-1}p)_{(B,i)} + o(1).$$

2. *Let* $k = k_B + k_R + k_G$*.*
$$\begin{aligned} \mathbf{E}\left(W_i(t+1) - W_i(t)\right) &= -k\frac{iW_i(t)}{E(t)} + o(1), \\ \mathbf{E}\left(B_i(t+1) - B_i(t)\right) &= k_B\frac{(i+1)(W_{i+1}(t) + B_{i+1}(t))}{E(t)} \\ &\quad -k\frac{iB_i(t)}{E(t)} - \frac{h(i)C_i(t)}{H(t)} + o(1). \end{aligned}$$

3. *There exists* $\rho > 0$ *such that*
$$\mathbf{Pr}\left[\textit{more than } s \textit{ copies are exposed in round } t\right] < (1-\rho)^s + o(1/n).$$

Sketch of proof. It is easy to show that if $\lambda_1(M) < 1$ then the probability that there exists a round exposing more than $(\log n)^2$ copies is $o(1)$. Hence, up to lower order terms, one can analyze an algorithm which always stops after having exposed $(\log n)^2$ copies in one round, even if there are 1-color-vertices remaining. Thus, for this new process the probabilities of generating vertices of a certain type do not shift by more than $O((\log n)^2/n)$ in one round, and one can easily find probability distributions dominating the analysis of the (modified) algorithm and also distributions dominated by this algorithm, all within $o(1)$ of the expected changes. Part (3) then follows from the previous observations of standard branching process theory applied to those distributions. Part (1) of the Lemma follows by recalling that $((I - M)^{-1}p)_{(B,i)}$ is the total expected progeny of type (B, i), and as such progenies have i copies, the statement follows. To prove the first equation of Part (2) observe the following, the number of 3-color-vertices of a certain degree i, $W_i(t)$, is non-increasing. It decreases if the second copy exposed in any iteration of the round t belongs to a 3-color-vertex of degree i. There are $iW_i(t)$ copies of this type, $E(t)$ unexposed copies in total, and since there are k copies exposed in expectation, the result follows. To prove the second equation of Part (2), we note that $B_i(t)$ increases if the 1-color-vertex or the 2-color-vertex (in the first iteration of the round) is colored B and matched with either a copy belonging to a 3-color-vertex of degree $i+1$ or with a copy of a 2-color-vertex having color list $\{G, R\}$ of $i+1$, yielding the first term. If the 1-color-vertex or the 2-color-vertex (in the first iteration of the round) is matched with a 2-color-vertex with color list $\{G, R\}$ and of degree i, then the number decreases, giving the second term. The third term counts for the fact that in the first iteration of the round the chosen 2-color-vertex might also be a vertex with color list $\{G, R\}$ and of degree i.

The equations in Part (2) of Lemma 3 suggest the application of the DEM. Let $z = (r_i, g_i, b_i, w_i)_{i=0}^{\Delta} \in \mathbb{R}^{4\times(\Delta+1)}$ be a vector of reals (with the intuition that they correspond to the uppercase letters scaled by n), and let $u^{(1)} = r^{(1)} + g^{(1)} + b^{(1)}$, where $s^{(q)} = \sum_i^q i s_i$, for a sequence $\{s_i\}$ and any integer $q \geq 1$. Analogously to M define the matrix A as follows: For $x \neq y$ we have

$$A_{(x,i),(y,j)} = \frac{j(i+1)c_{i+1}}{w^{(1)} + u^{(1)}},$$

where $c = \{r, g, b\} \setminus \{x, y\}$, and for $x = y$ it is 0. If $u^{(1)} > \alpha$ and $\lambda_1(A) < 1 - \beta$, then z is called (α, β)-subcritical. For $p_{(c,i)} = \frac{1}{2}\frac{h(i)x_i + y_i}{u^{(1)}}$ with $\{x, y\} = \{r, g, b\} \setminus \{c\}$, k_c for $c \in \{r, g, b\}$ and k defined as before, one thus has the following system of differential equations for the specific choice of $h(i) = i^{\rho}$,

$$\begin{aligned}
\frac{dw_i}{dt} &= -k\frac{iw_i}{w^{(1)} + u^{(1)}}, \\
\frac{dc_i}{dt} &= -k_c\frac{(i+1)(w_{i+1} + c_{i+1})}{w^{(1)} + c^{(1)}} - k\frac{ic_i}{w^{(1)} + u^{(1)}} - \frac{i^{\alpha}c_i}{u^{(\rho)}}.
\end{aligned}$$

For the application of Wormald's theorem, define the domain $D = D(\alpha, \beta)$ to be the set of all (α, β)-subcritical points in $\mathbb{R}^{4\times(\Delta+1)}$. Two hypotheses of Wormald's theorem are easy to check: If a point $(R_i(t)/n, G_i(t)/n, B_i(t)/n, W_i(t)/n)_{i=0}^{\Delta}$ is in D, then the expected changes are given by Part (2) of Lemma 3, and defining f_i to be the corresponding scaled changes, we see that they are within $\lambda \in o(1)$ of the expected changes. Moreover, by Part (3) of Lemma 3, a suitable value of β can be easily found. To prove the Lipschitz condition observe that in D, by definition, $u^{(1)} > \alpha$, and thus, there exists a Lipschitz constant for the term $1/(w^{(1)} + u^{(1)})$. Since $u^{(1)}$ is bounded away from 0, the same holds for $u^{(\rho)}$ for $\rho \geq 0$, and thus this term also satisfies a Lipschitz condition. It remains to prove that $(I - A)^{-1}p$ satisfies a Lipschitz condition. By definition, each coordinate of the vector p as well as the entries of A satisfy a Lipschitz condition. To finish the argument, observe that for any two points $z, z' \in D$, the matrices $A(z)$ and $A(z')$ have entries bounded by Δ and the largest eigenvalue bounded by $1-\beta$, and thus by the above observations of multitype branching processes, the corresponding distributions of total expected progenies of each type are close. Thus all conditions of Wormald's theorem are satisfied.

Since for fixed x, y, i each element $A_{(x,i),(y,j)}$ is proportional to j, A has rank 1, and by rewriting the differential equations, it can be shown that the only nonzero eigenvalue of A is $\lambda_1(A) = \frac{2}{3}\frac{u^{(2)}-u^{(1)}}{w^{(1)}+u^{(1)}}$. After some manipulations (see [3] for details), when taking the initial conditions $w_i = e^{-d}d^i/i! - 3\phi$ and $u_i = 3\phi$ into account (ϕ shorthand for $\sum_{i>\Delta} e^{-d}d^i/i!$), the differential equations can be solved numerically. It turns out that for $d = 4.0309$ and $\rho = 13$ it is always $\lambda_1(A) \leq 0.99909$, and by Wormald's theorem, a.a.s., for any $C \in \{R, G, B, W\}$ and all[2] steps $0 \leq t \leq T$,

$$C_i(t) = c_i(t/n)n + o(n).$$

The authors show that their method can also be used to prove that with positive probability a random 4-regular graph is 3-colorable.

The previous result by Achlioptas and Moore on the 3-colorability of 4-regular graphs is framed in an interesting sequence of developments about the colorability of random r-regular graphs. It is widely known that for $r \geq 2$ a random r-regular graph is a.a.s. not bipartite, and thus has chromatic number at least 3. Molloy and Reed showed that for a random 6-regular graph it is a.a.s. at least 4 (unpublished result). The basic ingredient of the proof was the first moment method: the authors show that the expected number of 3-colorings of a random regular graph converges to zero. Besides the previous result by Achlioptas and Moore in 2003, the same authors [4] subsequently showed that the chromatic number of a r-regular graph ($r \geq 3$) is a.a.s. k or $k+1$ or $k+2$, where k is the smallest integer such that $r < 2k \ln k$. They also showed that if furthermore $r > (2k-1)\ln k$, then a.a.s. the chromatic number is either $k+1$ or $k+2$. They also gave an upper bound on the probability that it is $k+2$, which showed that 5-regular graphs can be 4-colored with positive probability. This result was subsequently improved by Shi and Wormald [72], who showed that the chromatic number of a random r-regular graph is a.a.s. 3 for $r = 4$, 4 for $r = 6$, and either 3 or 4 for $r = 5$. They also showed that a.a.s. the chromatic number of a r-regular graph, for all other $r \leq 10$, is restricted to a range of two integers. Their proofs were algorithmic. Later, using the second moment method in a different way, the authors in [26] proved that a.a.s. random 5-regular graphs are 3-colorable. A related result was obtained by Nesetril and Wormald [69] who prove that the acyclic edge chromatic number of a random r-regular graph is a.a.s. equal to $+1$, improving a previous result of Alon, Sudakov, and Zaks [8].

[2] For technical reasons one needs an upper bound T on t as well - but coloring from this point on is easy.

8 Conclusions

In this survey we tried to present the DEM as a tool to get expectation together with concentration, for some suitable problems. We presented several cases of study, spanning different fields of computer science and discrete mathematics. Our main interest has been in the setting of the equations and we did not deal with the solution of those equations. In some of the considered problems there have been precise analytic solutions and in other cases the solutions are given using numerical tools, using Runge-Kutta and similar methods, for which there are nice implementations in most of the mathematical solver systems: *Maple*, *Mathematica*, etc. Most of the material in the present survey is self-contained, with the exception of the last two sections, where the reader needs a certain level of maturity in probability theory. We hope that the survey is useful to "expose" some readers to a powerful technique, although due to the conditions of applicability the DEM not always can be applied. If so, we will feel we accomplished our goal.

References

[1] D. Achlioptas. Setting 2 variables at a time yields a new lower bound for random 3-SAT. In *32nd ACM Symposium on Theory of Computing*, pages 28–37, 2000.

[2] D. Achlioptas. Lower bounds for random 3-SAT via differential equations. *Theoretical Computer Science*, 265(1-2):159–185, 2001.

[3] D. Achlioptas and C. Moore. Almost all graphs with average degree 4 are 3-colorable. *Journal Computer System Science*, 67(2):441–471, 2003.

[4] D. Achlioptas and C. Moore. The chromatic number of random regular graphs. In K. Jansen, S. Khanna, J. D. P. Rolim, and D. Ron, editors, *Proceedings of the 8th International Workshop on Randomization and Computation*, volume 3122 of *Lecture Notes in Computer Science*, pages 219–228. Springer, 2004.

[5] D. Achlioptas and F. Ricci-Tersenghi. Random formulas have frozen variables. *SIAM J. Comput.*, 39(1):260–280, 2009.

[6] D. Achlioptas and G. B. Sorkin. Optimal myopic algorithms for random 3-SAT. In *41th IEEE Symposium on Foundations of Computer Science*, pages 590–600, 2000.

[7] N. Alon and J. H. Spencer. *The probabilistic method.* Wiley-Interscience, New York, third edition, 2008.

[8] N. Alon, B. Sudakov, and A. Zaks. Acyclic edge colorings of graphs. *Journal of Graph Theory*, 37(3):157–167, 2001.

[9] T. Apostol. *Mathematical Analysis.* Addison Wesley, 1957.

[10] K. Azuma. Weighted sums of certain dependent random variables. *Tokuku Math. Journal*, 19:357–367, 1967.

[11] E. Bender and R. Canfield. The asymptotic number of non-negative integer matrices with given row and column sums. *Journal of Combinatorial Theory, Series A*, 24:296–307, 1978.

[12] S. Bezroukov, R. Elsasser, B. Monien, R. Preis, and J. Tillich. New spectral lower bounds on the bisection width of graphs. In U. Brandes and D. Wagner, editors, *26th. Graph-Theoretic Concepts in Computer Science*, volume 1928, pages 23–34. Springer, Lecture Notes in Computer Science, 2000.

[13] B. Bollobás. A probabilistic proof of an asymptotic formula for the number of labelled regular graphs. *European Journal of Combinatorics*, 1:311–316, 1980.

[14] B. Bollobás. The isoperimetric number of random regular graphs. *European Journal of Combinatorics*, 9:241–244, 1984.

[15] B. Bollobás. *Random Graphs (Second Edition).* Cambridge University Press, Cambridge, 2001.

[16] S. Boucheron, O. Bousquet, and G. Lugosi. Concentration inequalities. In O. Bousquet, U. Luxburg, and G. Rotsch, editors, *Advanced Lectures in Machine Learning*, pages 208–240. Springer, 2004.

[17] D. Brélaz. New methods to color the vertices of a graph. *Communications of the ACM*, 22(4):251–256, 1979.

[18] A. Z. Broder, A. M. Frieze, and E. Upfal. On the satisfiability and maximum satisfiability of random 3-CNF formulas. In *Proc. 4th ACM-SIAM Symposium on Discrete Algorithms*, pages 322–330, 1993.

[19] J. Cain and N. C. Wormald. Encores on cores. *Electronic Journal of Combinatorics*, 13(1), 2006.

[20] M.-T. Chao and J. V. Franco. Probabilistic analysis of two heuristics for the 3-satisfiability problem. *SIAM Journal of Computing*, 15(4):1106–1118, 1986.

[21] V. Chvátal and B. Reed. Mick gets some (the odds are on his side). In *33rd Annual Symposium on Foundations of Computer Science*, pages 620–627, 1992.

[22] C. Cooper, A. M. Frieze, and B. A. Reed. Random regular graphs of non-constant degree: Connectivity and hamiltonicity. *Combinatorics, Probability & Computing*, 11(3), 2002.

[23] F. den Hollander. *Large Deviations*, volume 14. American Mathematical Society, Philadelphia, 2000.

[24] J. Díaz, , M. J. Serna, and N. Wormald. Computation of the bisection for random d-regular graphs. *Theoretical Computer Science*, 382:120–130, 2007.

[25] J. Díaz, N. Do, M. Serna, and N. C. Wormald. Bounds on the max and min bisection of random cubic and 4-regular graphs. *Theoretical Computer Science*, 307:531–547, 2003.

[26] J. Díaz, A. Kaporis, G. Kemkes, L. Kirousis, X. Pérez, and N. C. Wormald. On the chromatic number of a random 5-regular graphs. *Journal of Graph Theory*, 61:157–191, 2009.

[27] J. Díaz, L. Kirousis, D. Mitsche, and X. Pérez. On the satisfiability threshold of formulae with three literals per clause. *Theoretical Computer Science*, 410:2725–2731, 2009.

[28] J. Díaz, J. Petit, and M. Serna. A guide to concentration bounds. In S. Rajasekaran, P. Pardalos, J. Reif, and J. Rolim, editors, *Handbook on Randomized Computing*, volume II, chapter 12, pages 457–507. Kluwer Press, New York, 2001.

[29] D. Dubhashi. Martingales and locality in distributed computing. In J. Rolim, M. Luby, and M. Serna, editors, *Randomization and Approximation Techniques in Computer Science*, volume 1518 of *Lecture Notes in Computer Science*, pages 60–70. Springer-Verlag, Berlin, 1998.

[30] D. P. Dubhashi and A. Panconesi. *Concentration of measure for the analysis of randomized algorithms*. Cambridge U.P., 2009.

[31] O. Dubois, Y. Boufkhad, and J. Mandler. Typical random 3-SAT formulae and the satisfiability threshold. *Electronic Colloquium on Computational Complexity (ECCC)*, 10(007), 2003.

[32] W. Duckworth and N. C. Wormald. Minimum independent dominating sets of random cubic graphs. *Random Structures and Algorithms*, 21(2):147–161, 2002.

[33] W. Duckworth and N. C. Wormald. On the independent domination number of random regular graphs. *Combinatorics, Probability & Computing*, 15(4):513–522, 2006.

[34] W. Duckworth and M. Zito. Large independent sets in random regular graphs. *Theoretical Computer Science*, 410(50):5236–5243, 2009.

[35] P. Erdős and A. Renyi. On the existence of a factor of degree one of a connected random graph. *Acta Mathematica of the Hungarian Academy of Sciences*, 17:359–366, 1966.

[36] W. Fernandez de la Vega. Random 2-SAT: results and problems. *Theoretical Computer Science*, 265(1-2):131–146, 2001.

[37] P. Flajolet, D. E. Knuth, and B. Pittel. The first cycles in an evolving graph. *Discrete Mathematics*, 75(1-3):167–215, 1989.

[38] E. Friedgut and J. Bourgain. Sharp thresholds of graph properties, and the k-SAT problem. *Journal of American Mathematical Society*, 12:1017–1054, 1999.

[39] A. M. Frieze and T. Łuczak. On the independence and chromatic numbers of random regular graphs. *Journal of Combinatorial Theory, Ser. B*, 54(1):123–132, 1992.

[40] A. M. Frieze and S. Suen. On the independence number of random cubic graphs. *Random Structure and Algorithms*, 5:649–664, 1994.

[41] A. M. Frieze and S. Suen. Analysis of two simple heuristics on a random instance of k-SAT. *Journal of Algorithms*, 20(2):312–355, 1996.

[42] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, San Francisco, 1979.

[43] A. Goerdt. A threshold for unsatisfiability. *Journal of Computer and System Sciences*, 53(3):469–486, 1996.

[44] G. Grimmett and D. S. (3rd Edition). *Probability and Random Processes.* Oxford University Press, Reading, Mass., 2001.

[45] T. Hagerup and C. Rub. A guided tour of Chernoff bounds. *Information Processing Letters*, 33:305–308, 1990.

[46] M. T. Hajiaghayi and G. Sorkin. The satisfiability threshold of random 3-SAT is at least 3.52. Technical report, IBM Research Report, 2003.

[47] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal American Statistics Association*, 58:13–30, 1963.

[48] S. Janson and M. J. Łuczak. A simple solution to the k-core problem. *Random Struct. Algorithms*, 30(1-2):50–62, 2007.

[49] S. Janson, T. Łuczak, and A. Rucinski. *Random Graphs.* Wiley-Interscience, New York, 2000.

[50] A. C. Kaporis, L. Kirousis, and E. G. Lalas. The probabilistic analysis of a greedy satisfiability algorithm. *Random Struct. Algorithms*, 28(4):444–480, 2006.

[51] R. M. Karp and M. Sipser. Maximum matchings in sparse random graphs. In *Proceedings of the 22nd IEEE Annual Symposium on Foundations of Computer Science*, pages 364–375, 1981.

[52] L. Kirousis, E. Kranakis, D. Krizanc, and Y. C. Stamatiou. Approximating the unsatisfiability threshold of random formulas. *Random Structures and Algorithms*, 12(3):253–269, 1998.

[53] A. Kostochka and L. Melnikov. On bounds of the bisection width of cubic graphs. In J. Nesetril and M. Fiedler, editors, *Fourth Czechoslovakian Symposium on Combinatorics, Graphs and Complexity*, pages 151–154. Elsevier Science Publishers, 1992.

[54] T. G. Kurtz. Solutions of ordinary differential equations as limits of pure Markov jump processes. *Journal of Applied Probability*, 7:49–58, 1970.

[55] T. G. Kurtz. *Approximation of Population Processes*, volume 36 of *CBMS-NSF Regional Conference Series in Applied Mathematics.* SIAM, 1981.

[56] T. Łuczak. Size and connectivity of the k-core of a random graph. *Discrete Mathematics*, 91(1):61–68, 1991.

[57] E. Maneva and A. Sinclair. On the satisfiability threshold and clustering of solutions of random 3-SAT formulas. *Theoretical Computer Science*, 407(1-3):359–369, 2008.

[58] C. McDiarmid. On the method of bounded differences. In J. Siemons, editor, *Surveys in Combinatorics, 1989*, pages 669–188. London Mathematical Society Lectures Notes Series, 1989.

[59] C. McDiarmid. Concentration. In H. Habib, C. McDiarmid, J. Ramirez, and B. Reed, editors, *Probabilistic Methods for Algorithmic Discrete Mathematics*, pages 195–248. Springer-Verlag, 1998.

[60] B. D. McKay. Independent sets in regular graphs of high girth. *Ars Combinatoria*, 23A:179–185, 1986.

[61] B. D. McKay and N. C. Wormald. Uniform generation of random regular graphs of moderate degree. *Journal of Algorithms*, 11:52–67, 1990.

[62] M. Mezard, G. Parisi, and R. Zecchina. Analytic and algorithmic solution of random satisfiability problems. *Science*, 297(812), 2002.

[63] M. Mezard and R. Zecchina. The random k-satisfiability problem: from an analytic solution to an efficient algorithm. *Physics Review*, E 66-056126, 2002.

[64] M. Mitzenmacher. Tight threshholds for the pure literal rule. Technical report, SRC Technical Note 1997-011, 1997.

[65] M. Mitzenmacher, A. Richa, and R. Sitaraman. The power of two random choices: A survey of techniques and results. In S. Rajasekaran, P. Pardalos, J. Reif, and J. Rolim, editors, *Handbook on Randomized Computing*, volume I, pages 255–312. Kluwer Press, New York, 2001.

[66] M. Mitzenmacher and E. Upfal. *Probability and Computing.* Cambridge U.P., 2005.

[67] M. Molloy and B. Reed. *Graph Coloring and the Probabilistic Method.* Springer, 2000.

[68] R. Motwani and P. Raghavan. *Randomized Algorithms.* Cambridge University Press, 1995.

[69] J. Nesetril and N. C. Wormald. The acyclic edge chromatic number of a random d-regular graph is $d + 1$. *Journal of Graph Theory*, 49(1):69–74, 2005.

[70] B. Pittel, J. Spencer, and N. C. Wormald. Sudden emergence of a giant-core in a random graph. *Journal of Combinatorial Theory, Ser. B*, 67:111–151, 1996.

[71] O. Riordan. The k-core and branching processes. *Combinatorics, Probability & Computing*, 17(1):111–136, 2008.

[72] L. Shi and N. C. Wormald. Colouring random 4-regular graphs. *Combinatorics, Probability & Computing*, 16:309–344, 2007.

[73] A. Steger and N. Wormald. Generating random regular graphs quickly. *Combinatorics, Probability & Computing*, 8:377–396, 1999.

[74] L. J. Stockmeyer and V. V. Vazirani. NP-completeness of some generalizations of the maximum matching problem. *Information Processing Letters*, 15(1):14–19, 1982.

[75] N. C. Wormald. Differential equations for random processes and random graphs. *Annals of Applied Probability*, 5:1217–1235, 1995.

[76] 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*, pages 73–155. PWN, Warsaw, 1999.

[77] N. C. Wormald. Models of random regular graphs. In D. P. J.D. Lamb, editor, *Surveys in Combinatorics*, volume 276 of *London Mahematical Society Lecture Notes Series*, pages 239–298. Cambridge University Press, 1999.

[78] N. C. Wormald. Analysis of greedy algorithms on graphs with bounded degrees. *Discrete Mathematics*, 273(1-3):235–260, 2003.

[79] M. Zito. Small maximal matchings in random graphs. *Theoretical Computer Science*, 297(1-3):487–507, 2003.