



Packing Vertices and Edges in Random Regular Graphs

Mihalis Beis

*Department of Computer Science
University of Liverpool
Liverpool, L69 3BX, UK*

William Duckworth

*Department of Computing
Macquarie University
Sydney, NSW 2109, Australia*

Michele Zito

*Department of Computer Science
University of Liverpool
Liverpool, L69 3BX, UK*

ABSTRACT

In this paper we consider the problem of finding large collections of vertices and edges satisfying particular separation properties in random regular graphs of degree r , for each fixed $r \geq 3$. We prove both constructive lower bounds and combinatorial upper bounds on the maximal sizes of these sets. The lower bounds are proved by analysing a class of algorithms that return feasible solutions for the given problems. The analysis uses the differential equation method proposed by Wormald [33]. The upper bounds are proved by direct combinatorial means. © ??? John Wiley & Sons, Inc.

D R A F T

August 17, 2006, 3:08pm

D R A F T

1. INTRODUCTION

A *regular graph* $G = (V, E)$ of *degree* r (or simply an r -regular graph) is a graph, all vertices of which have the same number r of incident edges. An r -regular graph on n vertices contains $rn/2$ edges therefore it is a requirement that rn must be even. The distance between two vertices in a graph is the number of edges in a shortest path between the two vertices. The distance between two edges $\{u_1, u_2\}$ and $\{v_1, v_2\}$ is the minimum of the distances between any two of the vertices u_i and v_j .

For any positive integer k , a k -*independent set* (resp. a k -*separated matching*) of a graph, is a set of vertices (edges), with the additional constraint that the minimum distance between any two vertices (edges) in the set is at least $k + 1$ (resp. k). Let $\alpha_k(G)$ (resp. $\nu_k(G)$) be the size of the largest k -independent sets (resp. k -matchings) in the graph G . For $j = 1$ (resp. $j = 2$) and any $k \geq 1$, the *maximum (k, j) -packing* ((k, j) PACKING) problem asks for a k -independent set (resp. k -matching) of size $\alpha_k(G)$ (resp. $\nu_k(G)$).

KNOWN RESULTS ON INDEPENDENT SETS.

Finding large k -independent sets has applications in the fields of job-scheduling on k -machines, VLSI design layout, routing and channel assignment location [19]. Many of these applications are in the field of distributed computing [23] and, as networks often have bounded or even regular degree, it is of interest to consider algorithms for finding large k -independent sets of such graphs. The $(1, 1)$ PACKING problem is the well known NP-hard problem of finding a maximum cardinality independent set of the given graph [17]. Kong and Zhao [21] showed that for every $k \geq 2$, $(k, 1)$ PACKING is NP-hard. They also showed that this problem remains NP-hard for regular bipartite graphs when $k \in \{2, 3, 4\}$ [22].

Due to the NP-hardness of the k -independent set problem, we are forced to relax the optimality requirement and consider heuristics that find a solution that is somehow close to optimal in a time that is bounded by a polynomial of the input size. Constant factor approximations exist for graphs of bounded maximum degree [6, 18]. Duckworth [10] presented a deterministic algorithm for finding a large 2-independent set in cubic (i.e. 3-regular) graphs. Analysing the performances of such algorithm it was shown that the size of a maximum 2-independent set of an n -vertex cubic graph is at least $\frac{n}{8} + O(1)$. The linear

programming technique that was used in the analysis also demonstrated the existence of an infinite family of cubic graphs for which the algorithm only achieves this bound. Note that for n -vertex r -regular graphs, it is simple to show that the size of a maximum 2-independent set is at most $\frac{n}{r+1}$ and at least $\frac{n}{r^2+1}$.

Simple heuristics often have a relatively poor worst-case performance (the interested reader may want to consult [4]) as there may exist many extremal input instances on which a simple algorithm may perform badly. It is therefore natural to consider the average-case performance of such heuristics. The maximum independent set problem has been studied thoroughly in the binomial random graph model (see [20]). Recently, results have appeared on the $(k, 1)$ PACKING problem, for $k > 1$ [3, 29]. Relatively less is known of α_k for (random) regular graphs. For the case $k = 1$, the current best known lower bounds on the size of a maximum independent set of random d -regular graphs are due to Wormald [33] and the current best known upper bounds are due to McKay [27]. Two of us investigated $(2, 1)$ PACKING in [12]. Assiyatun [2] gave an existence proof of a lower bound on the size of a largest 2-independent set of a random r -regular graph for $r \in \{3, 4, 5\}$. However, the analysis technique used there does not present an actual algorithm for finding a large 2-independent set.

KNOWN RESULTS ON MATCHINGS.

The $(1, 2)$ PACKING problem is the classical maximum matching problem. Stockmeyer and Vazirani [31] introduced the generalised $(k, 2)$ PACKING for $k \geq 2$, motivating it (for $k = 2$) as the “risk-free marriage problem” (find the maximum number of married couples such that each person is compatible only with the person (s)he is married to). The $(2, 2)$ PACKING problem (also known as the *maximum induced matching* problem) stimulated much interest in other areas of theoretical computer science and discrete mathematics as finding a maximum 2-matching of a graph is a sub-task of finding a strong edge-colouring of a graph (a proper colouring of the edges such that no edge is incident with more than one edge of the same colour as each other, see (for example) [14, 15, 25, 30]).

The $(k, 2)$ PACKING problem is NP-hard [31] for each $k \geq 2$ (polynomial time solvable [13] for $k = 1$). Improved complexity results are known for $(1, 2)$ PACKING [28] on random instances. In particular it has been proven that simple greedy heuristics produce sets of $\frac{n}{2} - o(n)$ independent edges [1] in dense random graphs and random regular graphs

with probability tending to one as n grows to infinity. A number of results are known on the approximability of an optimal 2-matching [8, 24, 35]. In particular the algorithm we present for 2-matchings has been analysed deterministically in [9] where it was shown to return a 2-matching of size at least $\frac{r(n-2)}{2(2r-1)(r-1)}$ in a connected r -regular graph on n vertices, for each $r \geq 3$. Furthermore, it was shown that there exist infinitely many r -regular graphs on n -vertices for which the algorithm only achieves this bound.

Zito [36] presented some simple results on the approximability of an optimal 2-matching in dense random graphs. For the case $r = 3$, the cardinality of a largest 2-matching \mathcal{M} of a random 3-regular graph a.a.s. satisfies $0.26645n \leq |\mathcal{M}| \leq 0.282069n$ [11] (unfortunately the optimistic $0.270413n$ lower bound claimed in the paper is not correct). Preliminary a.a.s. results on the $(k, 2)$ PACKING problem appeared in [5]. Finally, existential lower bounds on the size of the optimal 2-matching of random regular graphs are given in [2].

In this paper, we consider natural heuristics for approximating the solution to the problems defined above, and analyse their performance on random regular graphs. For $k \leq j$ our algorithms mimic a greedy process that, at each step, selects a vertex (resp an edge) of minimum positive degree, and adds it to the structure that is being built, removing from the graph all edges that are “close” to the chosen item. A particular feature of the problems considered when $k > j$ forces us to devise a slightly less immediate strategy to solve those cases. We also prove combinatorial upper bounds on $\alpha_k(G)$ and $\nu_k(G)$ using a direct expectation argument.

In the next section we present the model used for generating regular graphs u.a.r. (uniformly at random) along with an informal statement of the results proved in this paper and a description of the proof techniques used to prove them. In Section 3. we describe the class of randomised algorithms that we analyse. Section 4 presents the analysis of our algorithms, by first sketching the approach we use and then considering in turns each of the problems defined above. Finally Section 5 provides details about the upper bounds.

2. RANDOM GRAPH MODELS AND RESULTS

Let $\mathcal{G}(n, r\text{-reg})$ denote the uniform probability space of r -regular graphs on n vertices. Notation $G \in \mathcal{G}(n, r\text{-reg})$ will signify that G is selected according to such model.

A well known construction that gives the elements of $\mathcal{G}(n, r\text{-reg})$ is the *configuration model* (see, for example, [20, Chapter 9]). Let n urns be given, each containing r balls. A set F of $rn/2$ unordered pairs of balls is chosen u.a.r.. Let Ω be the set of all such pairings. Each $F \in \Omega$ corresponds to an r -regular (multi)graph with vertex set $V = \{1, \dots, n\}$ and edge set E formed by those sets $\{i, j\}$ for which there is at least one pair with one ball belonging to urn i and the other ball belonging to urn j . Let Ω^* be the set of all pairings not containing an edge joining balls from the same urn or two edges joining the same two urns. Since each simple graph corresponds to exactly $(r!)^n$ such pairings, a random pairing $F \in \Omega^*$ corresponds to an r -regular graph G without loops or multiple edges chosen u.a.r.

There are two features of this model that are particularly useful to our purposes. First, the model gives a basis for proving properties of random regular graphs by performing computations in Ω and conditioning on the event that the corresponding graph be simple since any event holding a.a.s. for a random r -regular multigraph also holds a.a.s. for a random graph in $\mathcal{G}(n, r\text{-reg})$. Second, notice that a random pairing can be picked by choosing pairs one after the other. Moreover, the first point in a pair may be selected using any rule whatsoever, as long as the second point is chosen u.a.r. from all the remaining free (unpaired) points. This property implies the existence of two equivalent ways of describing each of the algorithms presented in this paper. On one hand we can present them as working on a previously generated (random) graph, on the other we could consider a process that, at the same time, generates the graph and the structure of interest. Our exposition will be given in terms of the first type of description, but the second one will be used in our analysis.

In what follow we say that a property $\mathcal{B} = \mathcal{B}_n$ of a random graph holds asymptotically almost surely (a.a.s.) if the probability that \mathcal{B} holds tends to 1 as n tends to infinity. For other basic random graph theory definitions we refer the reader to [20].

In this paper we prove non-trivial lower and upper bounds on $\alpha_k(G)$ and $\nu_k(G)$ that hold a.a.s. assuming $G \in \mathcal{G}(n, r\text{-reg})$. The tables below describe the specific bounds for the first few values of r and k .

k	2		3		4		5	
r	l.b.	u.b.	l.b.	u.b.	l.b.	u.b.	l.b.	u.b.
3	0.204924	0.235551	0.090322	0.139057	0.048914	0.081725	0.022635	0.049812
4	0.142146	0.175705	0.040432	0.082791	0.01595	0.036443	0.004641	0.016137
5	0.106013	0.136759	0.021045	0.052816	0.006407	0.018165	0.001348	0.006136
6	0.082637	0.109913	0.01225	0.035646	0.002999	0.009956	0.000498	0.002688
7	0.066521	0.090557	0.007749	0.025172	0.001588	0.005888	0.000189	0.001318

Bounds on $\frac{\alpha_k(G)}{n}$.

k	1		2		3		4	
r	l.b.	u.b.	l.b.	u.b.	l.b.	u.b.	l.b.	u.b.
3	0.5	0.5	0.266454	0.282073	0.126406	0.156052	0.057922	0.094549
4	0.5	0.5	0.229526	0.25	0.079889	0.107573	0.023648	0.050068
5	0.5	0.5	0.204646	0.226949	0.055972	0.079217	0.011792	0.029335
6	0.5	0.5	0.18615	0.209101	0.041798	0.061096	0.006721	0.018586
7	0.5	0.5	0.171568	0.194651	0.032632	0.048756	0.004204	0.012506

Bounds on $\frac{\nu_k(G)}{n}$.

The proofs of our algorithmic results are based on the fact that, for each constant value of r and k , the algorithm dynamics can be described with sufficient precision by a random process that, for large n , behaves in a predictable way. Our analysis applies the differential equation method developed by Wormald (see e.g [33]). Following such approach we estimate the expected change for each of the variables defining the random process in question, and prove that, within a suitably defined domain, the variables satisfy a number of smoothness conditions. We are then able to apply a result in [34] to prove, essentially, that the variables stay close to their expected values.

3. THE ALGORITHMS

In this section we describe the greedy heuristics used to construct feasible solutions to instances of the various (k, j) PACKING problems. From now on a (k, j) -structure will be a k -independent set if $j = 1$ or a k -matching if $j = 2$. The specific case will be apparent from the context.

The algorithms are quite general and may be applied (with obvious modifications) to any graph. The analyses presented in Section 4. give lower bounds on the size of the resulting structures if the input graph is a random regular graph.

In what follows let $\Gamma(u) = \{v \in G : \{u, v\} \in E\}$ be the *neighbourhood of vertex u* . For each $i \in \{0, \dots, r\}$ let $V_i = V_i(G)$ denote the set of vertices whose neighbourhood contains i elements. Each of the algorithms that we consider fits the following description (an “element” here is either a vertex, for $j = 1$ or an edge, for $j = 2$)

Minimum Degree Process. While there is still edges in G , pick a vertex v of minimum positive degree, select a number of edges at distance at most $k + 1$ from v , add to \mathcal{S} an element incident to some of the selected edges, and remove all selected edges from G .

Here, \mathcal{S} denotes the (k, j) -structure returned by the algorithm. However, the analysis is greatly simplified if we consider a slightly more convoluted (but equivalent) class of algorithms. These algorithms will work in steps by repeatedly selecting and removing edges from the given graph G according to a set of rules $\text{Op}_1^{(k,j)}, \dots, \text{Op}_r^{(k,j)}$ which will be described in the following sections. Thus it is convenient to denote with G_t , for each integer t , the subgraph of the input graph G after t steps have been executed (here $G_0 = G$). With respect to the “twinned” pairing process we will denote by \mathcal{H}_t the collection of pairs selected in the first t steps of the process (conditioning on the final pairing being simple, such pairs correspond to the edges deleted from G during the first t steps). In all cases the resulting algorithm may be described by the following pseudo-code

Algorithm DegreeGreedy $_{k,j}(G)$

Input: a graph $G = (V, E)$ on n vertices.

$\mathcal{S} \leftarrow \emptyset;$

$t \leftarrow 0;$

```

while  $E \neq \emptyset$ 
  compute a probability distribution  $p_{(k,j)}(q, \frac{t}{n}, \frac{|V_1|}{n}, \dots, \frac{|V_r|}{n})$ ;
  update  $G$  and  $\mathcal{S}$  by performing  $\text{Op}_q^{(k,j)}$  with probability  $p_{(k,j)}(q, \frac{t}{n}, \frac{|V_1|}{n}, \dots, \frac{|V_r|}{n})$ ;
   $t \leftarrow t + 1$ ;
return  $\mathcal{S}$ .

```

Initially $V_r = V$ and $V_i = \emptyset$ for $i \leq r - 1$. The choice to perform $\text{Op}_q^{(k,j)}$, for $q \in \{1, \dots, r\}$ is based on a probability distribution $p_{(k,j)}(q, x, \mathbf{y})$. Such distribution, recomputed dynamically before a new operation is chosen, is affected by the birth of vertices of smaller and smaller degree which occur as the edges of G are successively removed. The general definition of $p_{(k,j)}(q, x, \mathbf{y})$ will be given in Section 4 (see (4.4)). An interesting property of the algorithms described by the pseudo-code above is that such definition only depends on $|V_1|, \dots, |V_r|$ and the particular moment in time, t . Furthermore, for each particular value of j and k , there will always be only at most $r - 1$ different distributions that will ever be used by the algorithm. Depending on the particular probability distribution $p_{(k,j)}(q, x, \mathbf{y})$ that is used at a given moment in time, the algorithm will be in one of a number of different *phases*. The outcome of our analysis implies that the algorithm processing, through successive phases, will mimic the minimum degree process described above.

The following sections contain a definition of the operations $\text{Op}_q(k, j)$ in each case, plus additional details that are specific to particular values of k and j .

Dense Matchings

A particularly simple heuristics may be used to find a large (k, j) -structure in a random r -regular graph, when¹ $k \leq j$. We focus on the case $j = 2$ as the analogue algorithms for $j = 1$ have been analysed before (see [16, 32]). For $q \in \{1, \dots, r\}$ let $\text{Op}_q^{(k,2)}$ denote the task of selecting a vertex v of degree q in a given graph, adding to \mathcal{S} an edge incident to v and to a vertex $u \in \Gamma(v)$ of minimum positive degree and then removing from the graph all edges at distance at most $k - 1$ from $\{u, v\}$.

¹we believe that the values reported in Section 1. justify the attribute "dense" in the title of this subsection.

Sparse Packings

Any obvious adaptation of the dense packing algorithm to the case $k > j$ fails. For $k = 2$ and $j = 1$, consider a subgraph of the input graph with vertex set $\{1, 2, 3, 4, 5, 6, 7\}$ and edge set $\{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{3, 5\}, \{4, 6\}, \{5, 7\}\}$. Choose vertex 1 to be part of the 2-independent set and delete all vertices at distance at most 2 from vertex 1. This leaves the edges $\{4, 6\}$ and $\{5, 7\}$ intact. Then choose vertex 4 (which is at distance 3 from vertex 1) to be part of the set. The algorithm deletes vertices 4 and 6 but no more as it has no knowledge that vertices 4 and 5 *were* connected by a path of length 2. It could then continue and pick vertex 5 (which is at distance 2 from vertex 4) to be part of the set. Since the problem is identical for matchings and independent sets, we resort to a different class of algorithms to solve the (k, j) PACKING problem when $k > j$. Such algorithms are based on the idea of repeatedly removing induced copies of a particular type of tree from the given graph. Let $t_0(r)$ be the trivial tree formed by a single vertex. Let $t_d(r)$ be the (rooted) tree obtained by taking r copies of $t_{d-1}(r)$ and joining their roots to a new vertex. For any integer $k \geq 2$, the tree $T_{k,j}(r)$ is a rooted tree whose root u_T has a child v which is the root of a copy of $t_{\lfloor k/2 \rfloor + j - 2}(r - 1)$ and $r - 1$ other children v_2, \dots, v_r which are roots of copies of $t_{\lfloor k/2 \rfloor - 1}(r - 1)$. In other words, $T_{k,1}(r)$ is formed by joining through a common root r identical complete $(r - 1)$ -ary trees, whereas a copy of $T_{k,2}(r)$ consists of two complete $(r - 1)$ -ary trees of depth $\lfloor \frac{k}{2} \rfloor$ whose roots are connected by an edge $e_T = \{u_T, v\}$. The algorithms used for $k > j$ will repeatedly try to find induced copies of $T_{k,j}(r)$, add either u_T or e_T to the set that is being built and remove all edges in $T_{k,j}(r)$ from the given graph. Of course a major difference w.r.t. the dense case is that the search for a copy of $T_{k,j}(r)$ may fail. In this case it is beneficial to stop the exploration immediately, remove any edge that has been probed in the failing attempt, and start a new attempt from scratch.

The description given so far is still too general as there are many possible ways in which an algorithm may search a graph for a copy of $T_{k,j}(r)$, and they are not all equivalent in terms of the cardinality of the structure returned. It turns out that the best alternative is to start exploring a possible copy of $T_{k,j}(r)$ from one of its leaves. For $q \in \{1, \dots, r\}$ let $\text{Op}_q^{(k,j)}$, denote the task of selecting a vertex v of degree q in a given graph, followed by an attempt to uncover $1 + (q - 1)(1 - ((k + j) \bmod 2))$ copies of $T_{k,j}(r)$ having v as a leaf.

For each complete copy of $T_{k,j}(r)$ that is found during an operation $\text{Op}_q^{(k,j)}$, an element (either u_T or e_T) is added to the structure that is being built. Finally all edges examined in this process (including those belonging to incomplete copies of $T_{k,j}(r)$ uncovered during a failing attempt) are removed from the graph. In particular, if $k + j$ is odd, distinct induced copies of $T_{k,j}(r)$ must also be vertex disjoint. Therefore all edges incident to the leaves of $T_{k,j}(r)$ must be removed as well.

4. ALGORITHMIC ANALYSIS

In order to obtain estimates on the size of the structures returned by the algorithms described in Section 3. we use the differential equation method proposed by Wormald (see e.g. [33]). In fact all our results rely on a refinement of his technique ([34, Theorem 1]). In this section we start by giving a summary of the general methodology (the interested reader is referred to [34] for a more detailed presentation). We then show in details how it can be applied in the context of induced matchings (or $(2, 2)$ PACKING). Finally, we will describe some of the calculations related to the other problems addressed by this work.

Given the input graph, all algorithms presented in this paper progress by peeling off a number of edges (upper bounded by an expression depending only on r , k , and j) from the graph and updating the structure \mathcal{S} (\mathcal{S}_t will denote the content of \mathcal{S} before G_t is further processed) that is being built. In each case we defined a set of elementary operations $\text{Op}_1^{(k,j)}, \dots, \text{Op}_r^{(k,j)}$ corresponding to each possible update (see Section 3.) and described each algorithm in terms of the operations allowed (with positive probability) at a particular moment in time.

The second nice property of the greedy algorithms considered in this paper is that, for each given value of k and j , we get good a.a.s. estimates on the size of the structure returned by the algorithm by analysing the random process $(Y_1(t), \dots, Y_{r+1}(t))$, where $Y_i(t) = |V_i(G_t)|$ for $i \in \{1, \dots, r\}$ and $Y_{r+1}(t) = |\mathcal{S}_t|$.

Assume we can define functions $f_{i,q}$ in \mathbb{R}^{r+2} such that the expected change to $Y_i(t)$, conditioned on \mathcal{H}_t and following one occurrence of $\text{Op}_q^{(k,j)}$ during step $t + 1$ is asymptotically $f_{i,q}(\frac{t}{n}, \frac{Y_1(t)}{n}, \dots, \frac{Y_{r+1}(t)}{n}) + o(1)$, for $i = 1, \dots, r + 1$, and any $q = 1, \dots, r$ such that $Y_q(t) > 0$. Furthermore assume these functions are continuous and bounded in

$$\mathcal{D}_\epsilon = \{(x, y_1, \dots, y_{r+1}) : 0 \leq x \leq r, 0 \leq y_i \leq r \text{ for } 1 \leq i \leq r+1, y_r \geq \epsilon\}$$

for some pre-chosen value of $\epsilon > 0$. We can then consider the following $r - 1$ distinct systems of differential equations

$$\frac{dy_i}{dx} = F(x, \mathbf{y}, i, s) \quad (4.1)$$

where

$$F(x, \mathbf{y}, i, s) = \frac{f_{r-s-1, r-s}(x, \mathbf{y})}{f_{r-s-1, r-s}(x, \mathbf{y}) - f_{r-s-1, r-s-1}(x, \mathbf{y})} f_{i, r-s-1}(x, \mathbf{y}) +$$

$$- \frac{f_{r-s-1, r-s-1}(x, \mathbf{y})}{f_{r-s-1, r-s}(x, \mathbf{y}) - f_{r-s-1, r-s-1}(x, \mathbf{y})} f_{i, r-s}(x, \mathbf{y})$$

for $s \in \{1, \dots, r-2\}$, and

$$F(x, \mathbf{y}, i, r-1) = f_{i, 1}(x, \mathbf{y}).$$

Under some obvious smoothness conditions on the functions $f_{i,q}$ (stated precisely in Theorem 4.1 below) each of the systems in (4.1), coupled with a suitably defined initial condition, admits a unique solution over an $[x_{s-1}, x_s]$ (for $s \in \{1, \dots, r-1\}$), where

$x_0 = 0$ and x_s is defined as the infimum of those $x > x_{s-1}$ for which at least one of the following holds:

(C1) $f_{r-s-1, r-s-1}(x, \mathbf{y}) \geq 0$ or

$$f_{r-s-1, r-s}(x, \mathbf{y}) - f_{r-s-1, r-s-1}(x, \mathbf{y}) \leq \epsilon \text{ and } s < r-1;$$

(C2) the component $r-s$ of the solution falls below zero or

(C3) the solution is outside \mathcal{D}_ϵ or ceases to exist. (4.2)

Let $\tilde{\mathbf{y}} = \tilde{\mathbf{y}}(x) = (\tilde{y}_1(x), \dots, \tilde{y}_{r+1}(x))$ be the function defined inductively as follows:

For each $i \in \{1, \dots, r+1\}$, $\tilde{y}_i(0) = \frac{Y_i(0)}{n}$. For $s \geq 1$, $\tilde{\mathbf{y}}$ is the solution to (4.1) over $[x_{s-1}, x_s]$, with initial condition $\mathbf{y}(x_{s-1}) = \tilde{\mathbf{y}}(x_{s-1})$. (4.3)

The following result (essentially a restatement of Theorem 1 in [34]) asserts that these functions describe the dynamics of an algorithm obtainable from the template $\text{DegreeGreedy}_{k,j}$ given in Section 3. by using particular sets of operations $\text{Op}_1^{(k,j)}, \dots, \text{Op}_r^{(k,j)}$, and $r-1$ different probability distributions $p^{(k,j)}(q, x, \mathbf{y})$ depending on the systems (4.1). In the

following statement expressions like $f'_{i,q}(\bar{x}, \tilde{\mathbf{y}}(\bar{x}))$ (resp. $f'_{i,q}(\bar{x}, \tilde{\mathbf{y}}(\bar{x}))^-$ or $f'_{i,q}(\bar{x}, \tilde{\mathbf{y}}(\bar{x}))^+$) refer to the derivative of $f_{i,q}$ taken with respect to x at the given point $(\bar{x}, \tilde{\mathbf{y}}(\bar{x}))$ (resp. refer to the left or right derivative at the given point).

Theorem 4.1. *Let $r \geq 3$. For $1 \leq i \leq r$ and $1 \leq q \leq r$ let $Y_i(t)$ and $f_{i,q}$ be the functions defined above. Assume furthermore that*

- (i) *there is an upper bound, depending only upon r , on the number of edges deleted, and on the number of elements added to \mathcal{S} during any one operation;*
- (ii) *the functions $f_{i,q}$ are rational functions of x and y_1, \dots, y_{r+1} with no pole in \mathcal{D}_ϵ ;*
- (iii) *there exist positive constants C_1, C_2 and C_3 such that for $1 \leq i < r$, everywhere on \mathcal{D}_ϵ , $f_{i,q} \geq C_1 y_{i+1} - C_2 y_i$ when $q \neq i$, and $f_{i,q} \leq C_3 y_{i+1}$ for all q .*

Then there exists a positive integer $m \leq r - 1$ such that,

$$f_{r-s-1, r-s-1}(x_{s-1}, \tilde{\mathbf{y}}(x_{s-1})) < 0 \text{ and}$$

$$f_{r-s-1, r-s}(x_{s-1}, \tilde{\mathbf{y}}(x_{s-1})) - f_{r-s-1, r-s-1}(x_{s-1}, \tilde{\mathbf{y}}(x_{s-1})) > \epsilon$$

for $s \in \{1, \dots, \min\{r-2, m\}\}$; furthermore

$$f_{r-1, r-1}(x_0, \tilde{\mathbf{y}}(x_0)) > 0$$

$$f_{r-s, r-s-1}(x_{s-1}, \tilde{\mathbf{y}}(x_{s-1}))^+ f'_{r-s-1, r-s}(x_{s-1}, \tilde{\mathbf{y}}(x_{s-1}))^{++} - f'_{r-s, r-s}(x_{s-1}, \tilde{\mathbf{y}}(x_{s-1}))^+ f_{r-s-1, r-s-1}(x_{s-1}, \tilde{\mathbf{y}}(x_{s-1}))^+ > 0,$$

for $s \in \{2, \dots, \min\{r-2, m\}\}$,

$$f'_{r-s, r-s}(x_{s-1}, \tilde{\mathbf{y}}(x_{s-1}))^- > 0,$$

for $s \in \{2, \dots, m\}$, and

$$f'_{1,1}(x_{r-2}, \tilde{\mathbf{y}}(x_{r-2}))^+ > 0$$

if $m = r - 1$. Furthermore there is a randomised algorithm for which a.a.s. there exists t such that $Y_i(t) = n\tilde{y}_i(x_m) + o(n)$ for $1 \leq i \leq r + 1$. Also, for each $s \in \{1, \dots, m\}$, $\tilde{y}_i(x) \equiv 0$ for $i \in \{1, \dots, r - s - 1\}$ if $x \in [x_{s-1}, x_s]$.

Theorem 4.1 will be used to analyse algorithm DegreeGreedy $_{k,j}(G)$ where the probability distributions described in Section 3. satisfy the following definition:

$$p_{(k,j)}(q, x, \mathbf{y}) = \begin{cases} -\frac{f_{r-s-1, r-s-1}(x, \mathbf{y})}{f_{r-s-1, r-s}(x, \mathbf{y}) - f_{r-s-1, r-s-1}(x, \mathbf{y})} & q = r - s \\ \frac{f_{r-s-1, r-s}(x, \mathbf{y})}{f_{r-s-1, r-s}(x, \mathbf{y}) - f_{r-s-1, r-s-1}(x, \mathbf{y})} & q = r - s - 1 \\ 0 & \text{otherwise} \end{cases} \quad (4.4)$$

when $x \in [x_{s-1}, x_s]$, for each $s \in \{1, \dots, m\}$. The interval $[x_{s-1}, x_s]$ represents phase s when, following a minimum degree strategy, the algorithm will a.a.s. perform either $\text{Op}_{r-s}^{(k,2)}$ or $\text{Op}_{r-s-1}^{(k,2)}$ all the time.

We can now state the result which bounds from below the size of the optima of the (k, j) PACKING problems. The values of $y_{r+1}(x_m)$ were found solving the various systems numerically using Maple's Runge-Kutta Fehlberg method. The resulting values, for the first few values of k and r , are given in the columns marked "l.b." in the tables in Section 2.. Details of its proof will be completed in the subsequent parts of this section.

Theorem 4.2. *Let r, j and k be positive integers with $r \geq 3, j \leq 2$. For $q \in \{1, \dots, r\}$, define $p_{(k,j)}(i, x, \mathbf{y})$ as in (4.4) where functions $f_{i,q}$, for each $i \in \{1, \dots, r+1\}$ are defined in (4.5) for $k \leq 2$ and $j = 2$, in (4.6) for all cases when $k > j$ and $k+j$ is even, and in (4.7) for all other cases. Let m be the integer associated with Y_i and $f_{i,q}$ in Theorem 4.1. The algorithm $\text{DegreeGreedy}_{k,j}(G)$ a.a.s. returns a structure of size $n\tilde{y}_{r+1}(x_m) + o(n)$ where functions $\tilde{y}_1, \dots, \tilde{y}_{r+1}$ are defined in (4.3) and x_0, \dots, x_m in (4.2) when $G \in \mathcal{G}(n, r\text{-reg})$.*

Proof. (Sketch) Hypothesis (i) of Theorem 4.1 is immediate since in any operation only the edges involving the selected element and its neighbours within a constant distance are deleted, and a bounded number of elements are added to \mathcal{S} . The functions $f_{i,q}$ satisfy (ii) because, in each case, the possible singularities satisfy $\sum i y_i = 0$ which defines a region outside \mathcal{D}_ϵ . Hypothesis (iii) follows again using $\sum i y_i \geq y_r \geq \epsilon$ and the boundedness of the functions \tilde{y}_i (which follows from the boundedness of \mathcal{D}_ϵ). Thus, defining $\tilde{\mathbf{y}}$ as in (4.3) we may solve the various systems of differential equations and find m , verifying the conditions on $f_{i,q}$ and its derivatives stated in Theorem 4.1 at the appropriate points of the computation. It turns out that these hold in the given domain, for each r , for sufficiently small $\epsilon > 0$. For such ϵ , the value of $\tilde{y}_{r+1}(x_m)$ may be computed numerically, and then by Theorem 4.1, this is the asymptotic value of $|\mathcal{S}|$ where \mathcal{S} is the set returned

by DegreeGreedy $_{k,j}(G)$. So the conclusion in each case is that a random r -regular graph a.a.s. has a k -independent set (or matching) of size at least $n\tilde{y}_{r+1}(x_m) + o(n)$.

A. Induced Matchings

The results proven in this section are identical to those reported, for $k \leq 2$ in [5]. However the analysis given here is simpler and it results in systems of differential equations that can be numerically solved much more quickly than those described in [5].

We analyse the algorithm DegreeGreedy $_{k,2}$, for $k \leq 2$ described in Section 3. and prove that hypotheses (ii) and (iii) of Theorem 4.1 are satisfied (hypothesis (i) is an obvious consequence of the algorithm definition).

In the remainder of this paper if $\mathcal{P}(\dots)$ is a logical expression (typically obtained by applying boolean connectives to simple relational operators on integers) then $[\mathcal{P}(\dots)]$ is a function that returns one (zero) if the logical expression evaluates to TRUE (resp. FALSE).

The probability of creating a vertex of degree $i - 1$ in the neighbourhood of a given vertex u when removing an edge e_u is asymptotically $P_i = \frac{iY_i}{\sum iY_i}$. In what follows S_a^b will denote the sum of all P_i 's for $a \leq i \leq b$ (with the convention that $S_a^b = 0$ if $a > b$). The expected change in Y_i due to the degree changes in $\Gamma(u)$ following the removal of e_u can be approximated by $-Q_i(0)$ where:

$$Q_i(0) = P_i - P_{i+1} \quad \text{with} \quad P_{r+1} = 0.$$

Using the same reasoning, if $e_u = \{u, v\}$ the expected change in Y_i due to the removal of e_u and of any other edge incident to v is asymptotically $-Q_i(1)$ where:

$$Q_i(1) = \sum_{z=1}^r P_z([i = z] + (z - 1)Q_i(0)).$$

For each $i, q \in \{1, \dots, r-1\}$, and $x \in \{1, \dots, r\}$, let

$$\chi_{q,x} = (S_x^r)^q - (S_{x+1}^r)^q$$

$$\phi_{i,x} = -[i = x] - (x-1)Q_i(k-1)$$

$$\beta_{x,d} = \frac{\binom{q}{d}(P_x)^d(S_{x+1}^r)^{q-d}}{\chi_{q,x}} \quad d \in \{1, \dots, q\}$$

$$\varepsilon_{x,d,m} = (q-d)\frac{P_m}{S_{x+1}^r} \quad d \in \{1, \dots, q\}, x < r$$

$$\gamma_{i,m} = -[i = m] + [k = 1 \wedge i = m-1] - [k = 2](m-1)Q_i(0) \\ m \in \{x+1, \dots, r\}, x < r$$

Lemma 4.3. *Let $r \geq 3$, and $k \leq 2$. Suppose $\text{Op}_q^{(k,2)}$ are defined as the updates related to algorithm DegreeGreedy $_{k,2}$, for $k \leq 2$. For each $q \in \{1, \dots, r\}$, conditioned on \mathcal{H}_t and $\text{Op}_q^{(k,2)}$, the expected change to $Y_i(t)$ is asymptotically*

$$f_{i,q} \left(\frac{t}{n}, \frac{Y_1(t)}{n}, \dots, \frac{Y_{r+1}(t)}{n} \right) = \begin{cases} -[i = q] + \sum_{x=1}^r \left(\chi_{q,x}(\phi_{i,x} - \gamma_{i,x}) + \gamma_{i,x}qP_x(S_x^r)^{q-1} + \right. \\ \quad \left. + [x < r] \frac{q(\chi_{q,x} - P_x(S_x^r)^{q-1})}{S_{x+1}^r} \sum_{m=x+1}^r P_m \gamma_{i,m} \right), & i \leq r \\ 1 & i = r+1. \end{cases} \quad (4.5)$$

Proof. We calculate the expected change in Y_i when performing an operation $\text{Op}_q^{(k,2)}$ starting from a vertex u given \mathcal{H}_t by conditioning on the minimum degree of a vertex in $\Gamma(u)$ and then on the number of vertices of minimum degree in $\Gamma(u)$.

The probability that the minimum degree in $\Gamma(u)$ is x , is $\chi_{q,x} + o(1)$. Conditioned to this event, the expected change in $Y_i(t)$ due to the removal of all edges incident with the chosen minimum degree vertex $v \in \Gamma(u)$ and, for $k = 2$, all remaining edges incident to vertices in $\Gamma(v)$, is $\phi_{i,x} + o(1)$. To get an expression for $f_{i,q}(x, \mathbf{y})$ it is useful to further condition on the size of $V_x \cap \Gamma(u)$. The probability that $|V_x \cap \Gamma(u)| = d$ (where $1 \leq d \leq q$) given that the minimum degree of the vertices in $\Gamma(u)$ is x , is $\beta_{x,d} + o(1)$. The expected change

in Y_i due to the removal of all edges incident with the $d - 1$ vertices in $V_x \cap \Gamma(u) \setminus \{v\}$, conditioned on the minimum degree in $\Gamma(u)$ being x , is $(d - 1)\gamma_{i,x} + o(1)$. Finally, the expected size of $V_m \cap \Gamma(u)$ (where $x + 1 \leq m \leq r$) given that the minimum degree in $\Gamma(u)$ is x and $|V_x \cap \Gamma(u)| = d$, is $\varepsilon_{x,d,m} + o(1)$, with the convention that the expected value is zero if $x = r$. Putting all this together, $f_{i,q}(\frac{t}{n}, \frac{Y_1}{n}, \dots, \frac{Y_{r+1}}{n})$, for $i, q \in \{1, \dots, r\}$ can be written as

$$-[i = q] + \sum_{x=1}^r \chi_{q,x} \left(\phi_{i,x} + \sum_{d=1}^q \beta_{x,d} \left((d-1)\gamma_{i,x} + [x < r] \sum_{m=x+1}^r \varepsilon_{x,d,m} \gamma_{i,m} \right) \right)$$

(whereas $f_{r+1,q}(\frac{t}{n}, \frac{Y_1}{n}, \dots, \frac{Y_{r+1}}{n}) = 1$ since an edge is added to \mathcal{S} following each $\text{Op}_q^{(k,2)}$).

Distributing $\chi_{q,x}$ inside the main bracket, and replacing $\beta_{x,d}$ with its definition, the sum becomes

$$\sum_{x=1}^r \left(\chi_{q,x} \phi_{i,x} + \sum_{d=1}^q \binom{q}{d} (P_x)^d (S_{x+1}^r)^{q-d} \left((d-1)\gamma_{i,x} + [x < r] \sum_{m=x+1}^r \varepsilon_{x,d,m} \gamma_{i,m} \right) \right).$$

Since $\gamma_{i,x}$ does not depend on d , this can be written as

$$\sum_{x=1}^r \left(\chi_{q,x} \phi_{i,x} + \gamma_{i,x} (qP_x (S_x^r)^{q-1} - \chi_{q,x}) + [x < r] \sum_{d=1}^q \binom{q}{d} (P_x)^d (S_{x+1}^r)^{q-d} \sum_{m=x+1}^r \varepsilon_{x,d,m} \gamma_{i,m} \right)$$

Using the definition of ε such expression becomes

$$\begin{aligned} & \sum_{x=1}^r \left(\chi_{q,x} \phi_{i,x} + \gamma_{i,x} [qP_x (S_x^r)^{q-1} - \chi_{q,x}] + \right. \\ & \quad \left. + [x < r] \sum_{d=1}^q \binom{q}{d} (P_x)^d (S_{x+1}^r)^{q-d-1} (q-d) \sum_{m=x+1}^r P_m \gamma_{i,m} \right). \end{aligned}$$

Some further simplification is possible since the rightmost sum does not depend on d , giving

$$\sum_{x=1}^r \left(\chi_{q,x} \phi_{i,x} + \gamma_{i,x} [qP_x (S_x^r)^{q-1} - \chi_{q,x}] + [x < r] \frac{q(\chi_x - P_x (S_x^r)^{q-1})}{S_{x+1}^r} \sum_{m=x+1}^r P_m \gamma_{i,m} \right).$$

Notice that, for each i and q , $f_{i,q}(x, \mathbf{y})$ can be written as $(\sum_{i=1}^r i y_i)^{-(k+q)} \times p(\mathbf{y})$ where p is a polynomial of degree $k + q$ in y_1, \dots, y_{r+1} . Hypothesis (ii) of Theorem 4.1 therefore follows from the definition of \mathcal{D}_ε . Hypothesis (iii) can also be easily verified in each case. For instance, when $k = 2$ and $r = 3$,

$$\begin{aligned} f_{2,2}(x, y_1, y_2, y_3, y_4) = \\ -1 + \phi_{2,2} \chi_{2,2} + \phi_{2,3} \chi_{2,3} + \gamma_{2,2} P_2 (2P_1 + P_2) + \gamma_{2,3} P_3 (2P_1 + 2P_2 + P_3) \end{aligned}$$

which never exceeds some sufficiently large positive constant C . Thus, for a sufficiently small $\epsilon > 0$,

$$f_{2,2}(x, y_1, y_2, y_3, y_4) \leq \frac{C}{\epsilon} y_3.$$

B. Sparse case when $k + j$ is even

An operation $\text{Op}_q^{(k,j)}$ consists of the selection of a random vertex of degree q followed by an attempt to find q copies of $T_{k,j}(r)$ around it. It is convenient to talk of the vertices in $T_{k,j}(r)$ as separated into a number of levels. Level 0 is formed by a single leaf, level 1 by a vertex of degree r , level 2 by at least one vertex of degree r and $r - 2$ leaves. Generally level l (for $0 < l < 2\lfloor \frac{k-j}{2} + 1 \rfloor + j - 1$) is composed of $(r - 1)^{\lceil l/2 \rceil - 1}$ vertices of degree r and, when $l > 0$ is even, of $(r - 2)(r - 1)^{l/2 - 1}$ leaves. Level $2\lfloor \frac{k-j}{2} + 1 \rfloor + j - 1$ is composed of $(r - 1)^{\lfloor \frac{k-j}{2} + 1 \rfloor}$ leaves only.

Lemma 4.4. *Let $r \geq 3$ and $k > j$. Suppose that, for $q \in \{1, \dots, r\}$, $\text{Op}_q^{(k,j)}$ are defined as the updates related to algorithm DegreeGreedy, for $k > j$, when $k + j$ is even. For each $q \in \{1, \dots, r\}$, the expected change to $Y_i(t)$, conditioned on \mathcal{H}_t and $\text{Op}_q^{(k,j)}$, is asymptotically*

$$f_{i,q} \left(\frac{t}{n}, \frac{Y_1(t)}{n}, \dots, \frac{Y_{r+1}(t)}{n} \right) = \begin{cases} -[i = q] - q \left(Q_i(0) + \sum_{l=0}^{k-1} \prod_{m=0}^l \mathcal{P}_m \left((r-1)^{\lceil l/2 \rceil} [i = r-1] + (r-1)^{\lceil l/2 \rceil + 1} Q_i(0) \right) \right) & i \leq r \\ q \prod_{m=0}^{k-1} \mathcal{P}_m & i = r+1 \end{cases} \quad (4.6)$$

where \mathcal{P}_m represents the probability of succeeding at level m .

Proof. Success at level m occurs if the right combination of vertex degrees is found at level $m + 1$. Hence \mathcal{P}_m is asymptotically equal to $(P_r)^{(r-1)^{m/2}}$ when m is even, and it is $1 - (1 - \mathcal{P}_{m-1})^{r-1}$ otherwise. If success does occur at level l then the previously accounted for contribution to $f_{r-1,q}$ given by the removal of a single edge incident to each of the $(r - 1)^{\lceil l/2 \rceil}$ vertices of degree r at level $l + 1$ must be detracted, and then all

edges connecting vertices at level $l + 1$ with those at level $l + 2$ can be removed. This is asymptotically equal to $-(r - 1)^{\lfloor l/2 \rfloor} [i = r - 1] - (r - 1)^{\lfloor l/2 \rfloor + 1} Q_i(0)$.

Similarly the asymptotic expression for $f_{r+1,q}$ is $q \prod_{m=0}^{k-1} \mathcal{P}_m$ as q attempts are made to increase the size of \mathcal{S} during an $\text{Op}_q^{(k,j)}$ operation.

C. Sparse case for $k + j$ is odd

In this case an operation $\text{Op}_q^{(k,j)}$ consists of the selection of a random vertex v of degree q followed by an attempt to reveal a $T_{k,j}(r)$ structure around it. The analysis for this case is complicated by the requirement that distinct copies of $T_{k,j}(r)$ must be vertex disjoint. The expected change in Y_i can be computed, as in the case $k \leq 2$, by conditioning on the degree distribution in $\Gamma(v)$ but major differences arise. First of all we must condition on the maximum degree in $\Gamma(v)$ and some interesting updates occur only if this maximum degree is r . Secondly this conditioning needs to be performed at successive levels in the retrieval of a copy of $T_{k,j}(r)$ (otherwise the current trial has no hope of finding a copy of $T_{k,j}(r)$). Finally the major complication in the asymptotic expression for $f_{i,q}$ comes from the need of being able to delete all edges incident with the leaves of $T_{k,j}(r)$ and these leaves (in the graph) can have arbitrary degree.

Let c be an integer. For each $i, q \in \{1, \dots, r - 1\}$, and $x \in \{1, \dots, r\}$, let

$$\chi_x^c = (S_1^x)^{(r-1)^c} - (S_1^{x-1})^{(r-1)^c}$$

$$\beta_{x,d}^c = \binom{(r-1)^c}{d} (P_x)^d (S_1^{x-1})^{(r-1)^c - d} \quad d \in \{1, \dots, (r-1)^c\}$$

$$\varepsilon_{x,d,m}^c = ((r-1)^c - d) \frac{P_m}{S_1^{x-1}} \quad d \in \{1, \dots, (r-1)^c\}, x > 1$$

$$\zeta_{i,x,m} = -[i = m] + [x \neq r \wedge i = m - 1] - [x = r](m - 1)Q_i(0) \quad m \in \{1, \dots, r\},$$

Lemma 4.5. *Let $r \geq 3$ and $k > j$. Suppose that, for $q \in \{1, \dots, r\}$, $\text{Op}_q^{(k,j)}$ are defined as the updates related to algorithm DegreeGreedy, for $k > j$ when $k + j$ is odd. For each $q \in \{1, \dots, r\}$, the expected change to $Y_i(t)$, conditioned on \mathcal{H}_t and $\text{Op}_q^{(k,j)}$, is*

asymptotically

$$f_{i,q} \left(\frac{t}{n}, \frac{Y_1(t)}{n}, \dots, \frac{Y_{r+1}(t)}{n} \right) = \begin{cases} -[i = q] - qQ_i(0) + \\ \quad (1 - (1 - P_r)^q) \left(-[i = r - 1] + \right. \\ \quad \left. \sum_{x=1}^r \chi_x^1 (-[i = x] + [x \neq r \wedge i = x - 1] + [x = r] \Xi^{\lfloor \frac{k}{2} \rfloor + j - 2}(1) - \zeta_{i,x,x}) \right) & i \leq r \\ \zeta_{i,x,x} ((r - 1)P_x(S_1^x)^{r-2} - \chi_x^1) + [x > q] \frac{(r - 1)(\chi_x^1 - P_x(S_1^x)^{r-2})}{S_1^{x-1}} \sum_{m=1}^{x-1} P_m \zeta_{i,x,m} \\ \\ (1 - (1 - P_r)^q) \times \\ \quad P_0(1 + [j = 1]((P_r)^{(r-1)^{\frac{k}{2}-1}} - 1)) \prod_{l=1}^{\lfloor \frac{k}{2} \rfloor + j - 3} P_r^{(r-1)^l} \mathcal{P}_l. & i = r + 1 \end{cases} \quad (4.7)$$

In the expressions above \mathcal{P}_l stands for the probability of succeeding at a certain level in the discovery of a copy of $T_{k,j}(r)$. Furthermore $\Xi^b(a)$ describes the updating of the graph due to the inspection of all edges to vertices at level $2a$ and beyond in a candidate copy of $T_{k,j}(r)$.

Proof. Suppose that, for $c \geq 1$, $(r - 1)^{c-1}$ vertices of degree $r - 1$ are chosen in G_t . If c is independent of n then a.a.s. such vertices are adjacent to $(r - 1)^c$ distinct vertices. The probability that the maximum degree among these vertices be x , with $1 \leq x \leq r$ is $\chi_x^c + o(1)$. The probability of having exactly d vertices of degree x in the experiment outlined above is $\beta_{x,d}^c + o(1)$, while the expected number of vertices of degree m , with $1 \leq m < x$, is $\varepsilon_{x,d,m}^c + o(1)$. The expected change in $Y_i(t)$ due to the removal of all remaining edges out of a vertex of (initially) degree m is $\zeta_{i,x,m} + o(1)$. To complete the description of $f_{i,q}(x, \mathbf{y})$ we need to understand the meaning of the term $\Xi^{\lfloor \frac{k}{2} \rfloor + j - 2}(1)$. Function $\Xi^b(a)$ models the behaviour of the algorithm from level $2a$ onwards. Assume, generally, that the algorithm has reached an even level where there are $(r - 1)^{a-1}$ vertices of degree r , we remove the $r - 1$ edges incident with each one of them and we change the degree of $(r - 1)^a$ vertices which all must have degree r initially. This happens with probability $P_r^{(r-1)^a}$. Then we expose the remaining $r - 1$ edges from all of them and we have a success if there is at least 1, out of the possible $r - 1$, $(r - 1)^a$ -tuple being

composed of vertices with degree r . The success probability \mathcal{P}_a can be approximated as $1 - (1 - P_r^{(r-1)^a})^{r-1}$.

Since we condition on the maximum degree (see χ_x^c), when $x \neq r$ we certainly have a failure. If $x = r$ the success or failure depends on the number of vertices of degree r and in some cases on the arrangement of such vertices. Let d be the number of vertices of degree r out of the $(r-1)^{a+1}$ edges that have been exposed. Clearly, when $d < (r-1)^a$ we have a failure with probability 1. On the other hand, when $d \geq (r-1)^a + (r-2)((r-1)^a - 1)$ the success is certain. In any other case we may have either a failure or a success.

In case of a failure the expected behaviour of the algorithm can be described by

$$A_a = d(-[i = r] + [i = r - 1]) + \sum_{m=1}^{r-1} \varepsilon_{r,d,m}^{a+1} (-[i = m] + [i = m - 1]).$$

Let $(r-1)^a \leq d < (r-1)^a + (r-2)((r-1)^a - 1)$, for d fixed there are $\Lambda = \binom{(r-1)^{a+1}}{d}$ equiprobable cases for the arrangements of the vertices of degree r . We can choose $(r-1)^a$ vertices of degree r in $r-1$ ways in order to have a success and the remaining $d - (r-1)^a$ vertices of degree r may be distributed in any of the $\binom{(r-2)(r-1)^a}{d - (r-1)^a}$ possible ways. Hence there are $\Theta = (r-1) \binom{(r-2)(r-1)^a}{d - (r-1)^a}$ successful cases and $\Lambda - \Theta$ unsuccessful ones. Of course in case of a success the algorithm may proceed at the next even level where it either starts the above procedure all over again and this can be described by

$$\begin{aligned} \Xi^b(a) = & -(r-1)^a Q_i(0) + P_r^{(r-1)^a} \left(-[i = r - 1](r-1)^a + \right. \\ & \sum_{x=1}^{r-1} \left(([i = x - 1] - [i = x])((r-1)^{a+1} P_x(S_1^x)^{(r-1)^{a+1} - 1} - \chi_x^{a+1}) + \right. \\ & \left. \left. + [x > 1] \frac{(r-1)^{a+1} (\chi_x^{a+1} - P_x(S_1^x)^{(r-1)^{a+1} - 1})}{S_1^{x-1}} \sum_{m=1}^{x-1} \varepsilon_{x,d,m}^{a+1} ([i = m - 1] - [i = m]) \right) + \right. \\ & \left. \sum_{d=1}^{(r-1)^a - 1} \beta_{r,d}^{a+1} A_a + \sum_{d=(r-1)^a}^{(r-1)^a + (r-2)((r-1)^a - 1) - 1} \beta_{r,d}^{a+1} B_a + \sum_{d=(r-1)^a + (r-2)((r-1)^a - 1)}^{(r-1)^{a+1}} \beta_{r,d}^{a+1} (\Xi^b(a+1) + \xi_{i,d}^a) \right), \end{aligned}$$

with

$$\xi_{i,d}^a = -(r-1)^a [i = r] + (d - (r-1)^a) \zeta_{i,r,r} + \sum_{m=1}^{r-1} \varepsilon_{r,d,m}^{a+1} \zeta_{i,r,m},$$

or it has reached the final level which can be described by the following base case

$$\Xi^b(b+1) = -(r-1)^b \left(Q_i(j-1) + [j=1](P_r)^{(r-1)^b}([i=r-1] + (r-1)Q_i(j)) \right).$$

The behaviour of the algorithm in the cases where we may have either success or failure can be described by

$$B_a = \frac{\Theta}{\Lambda} (\Xi^b(a+1) + \xi_{i,d}^a) + \left(1 - \frac{\Theta}{\Lambda}\right) A_a.$$

5. UPPER BOUNDS

The argument leading to the upper bounds given in Section 2. is based on finding a close estimate on the expected number X_y of (k, j) -structures of size $y = \mu n$ for $G \in \mathcal{G}(n, r\text{-reg})$ in terms of a decreasing function of the form $f(\mu)^n$, for fixed values of k, j , and r , and then on the numerical estimation of the smallest value μ^* larger than the lower bounds found through the algorithms in Section 3. that makes $f(\mu) < 1$.

We focus on the case $k > j$ as the dense cases have been considered before [7, 11, 27] (notice in particular that our analysis covers, the case $k = j = 2$ for arbitrary values of r). For each value of k and j , since random regular graphs do not contain many short cycles, the expectation of X_y can be computed by counting occurrences of $T_{k,j}(r)$ in G . Furthermore standard results on the configuration model imply that such counting can indeed be performed on the analogous structures that can be identified in the underlying pairings. For the forthcoming exposition it is convenient to think of the vertices in each copy of $T_{k,j}(r)$ as arranged in levels u_T (resp. the end-points of e_T) being at level zero and its (their) descendants being at level i if their distance from u_T (resp. one of the end-points of e_T) is i . Let $\lambda = \lambda(k, j) = \lfloor \frac{k+j-2}{2} \rfloor - j$. In a (k, j) -structure of size y there are jy vertices at level 0, and in general $(r-j+1)jy(r-1)^i$ vertices at level i . Define $V(-1) = jy$ and, for $i \geq 0$,

$$V(i) = jy + (r-j+1)jy \frac{(r-1)^{i+1} - 1}{r-2}.$$

Furthermore, define

$$A_{k,j}(y) = \begin{cases} \binom{n}{y} & j = 1 \\ \binom{n}{2y} (2y-1)!! r^{2y} & j = 2 \end{cases}$$

and

$$R_{k,j}(y) = \begin{cases} (n-jy)_{(r-j+1)jy(r-1)\frac{(r-1)^{\lambda+1}-1}{r-2}} r^{(r-j+1)jy(r-1)\frac{(r-1)^{\lambda+1}-1}{r-2}} & k+j \text{ odd} \\ R_{k-1,j}(y)(r(n-V(\lambda(k,j))))_{(r-j+1)jy(r-1)^{\lambda+1}} & k+j \text{ even} \end{cases}$$

(with $R_{1,2}(y) = 1$). Finally, define

$$\Upsilon_{k,j}(y) = r(n - V(\lambda(k,j))) - (r-j+1)jy(r-1)^{\lambda(k,j)+1}$$

(notice that for each $h > 1$, $\Upsilon_{2h-1,1}(y) = \Upsilon_{2h,1}(y)$ whereas $\Upsilon_{2h,2}(y) = \Upsilon_{2h+1,2}(y)$).

Lemma 5.1. $E(X_y) \sim A_{k,j}(y)R_{k,j}(y)\frac{(\Upsilon_{k,j}(y)-1)!!}{(rn-1)!!}$, where $x!! = x(x-2)\dots 3\cdot 1$ for any odd positive integer x .

Proof. Calculations are performed on configurations. The term $A_{k,j}(y)$ counts the number of ways in which the y components of \mathcal{S} can be chosen. $R_{k,j}(y)$ counts the number of ways in which the elements in such structure can be embedded in a configuration so that they are at distance at least k from each other. Finally $\Upsilon_{k,j}$ counts the number of points still to be paired after the pairs associated with \mathcal{S} have been selected. Hence $A_{k,j}(y)R_{k,j}(y)(\Upsilon_{k,j}(y)-1)!!$ counts the number of configurations containing a (k,j) -structure of size y formed by copies of $T_{k,j}(r)$. Since structures formed in a different way are rare, the expectation of X_y can be computed by simply dividing such number by the total number of configurations on nr points.

Using Lemma 5.1 and Stirling's approximation to the factorial it is possible to prove that the asymptotic expression for $E(X_y)$ has the form $n^{O(1)}(f(\mu))^n$, with the function f being continuous and unimodal, greater than one for values of μ close to (but larger than) the lower bounds given in Section 2. and smaller than one for some $\mu < \frac{1}{2}$. We are now ready to state our main result.

Theorem 5.2. *Let $r \geq 3$. For each k and j , the value μ^* obtained by solving $f(\mu) = 1$ is such that a.a.s. $G \in \mathcal{G}(n, r\text{-reg})$ does not contain any (k, j) -structure of size greater than μ^*n .*

Maximal vs. non-maximal (k, j) -structures.

It should be remarked that slightly smaller upper bounds can be found by counting *maximal* (k, j) -structures (a stronger 0.28206915 value for $j = k = 2$ is reported in [11]). However we prefer to keep the simpler exposition presented above as the magnitude of the improvements makes the more complicated analysis rather uninteresting.

REFERENCES

- [1] J. Aronson, A. Frieze, and B. G. Pittel. Maximum matchings in sparse random graphs: Karp-Sipser revisited. *Random Structures and Algorithms*, 12:111–178, 1998.
- [2] H. Assiyatun. *Large Subgraphs of Regular Graphs*. Doctoral Thesis, Department of Mathematics & Statistics, The University of Melbourne, Australia, 2001.
- [3] G. Atkinson, A. Frieze. On the b -independence number of sparse random graphs. *Combinatorics, Probability and Computing*, 13:295–310, 2004.
- [4] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, M. Protasi. *Complexity and Approximation*, Springer Verlag, 1999.
- [5] M. Beis, W. Duckworth, M. Zito. Packing edges in random regular graphs. In K. Diks, W. Rytter, editors, *Mathematical Foundations of Computer Science 2002*, volume 2420 of *Lecture Notes in Computer Science*, pages 118–130. Springer-Verlag, 2002.
- [6] P. Berman, T. Fujito. On Approximation Properties of the Independent Set Problem for Low Degree Graphs. *Theory of Computing Systems*, 32:115–132, 1999.
- [7] B. Bollobás, B. D. McKay. The number of matchings in random regular graphs and bipartite graphs. *Journal of Combinatorial Theory*, B 41:80–91, 1986.
- [8] K. Cameron. Induced matchings. *Discrete Applied Mathematics*, 24(1-3):97–102, 1989.
- [9] W. Duckworth, D. Manlove, M. Zito. On the approximability of the maximum induced matching problem. *Journal of Discrete Algorithms*, 3:79–91, 2005.

- [10] W. Duckworth. *Greedy Algorithms and Cubic Graphs*. Doctoral Thesis, Department of Mathematics & Statistics, The University of Melbourne, Australia, 2001.
- [11] W. Duckworth, N. C. Wormald, M. Zito. Maximum induced matchings of random cubic graphs. *Journal of Computational and Applied Mathematics*, 142(1):39–50, 2002.
- [12] W. Duckworth, M. Zito. Large 2-independent sets of regular graphs. *Electronic Notes in Theoretical Computer Science*, 91(C):43–55, 2004.
- [13] J. Edmonds. Paths, trees and flowers. *Canadian Journal of Mathematics*, 15:449–467, 1965.
- [14] P. Erdős. Problems and results in combinatorial analysis and graph theory. *Discrete Mathematics*, 72:81–92, 1988.
- [15] R. J. Faudree, A. Gyárfas, R. H. Schelp, Z. Tuza. Induced matchings in bipartite graphs. *Discrete Mathematics*, 78(1-2):83–87, 1989.
- [16] A.M. Frieze, S. Suen. On the Independence Number of Random Cubic Graphs. *Random Structures and Algorithms*, 5:649–664, 1994.
- [17] M.R. Garey, D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman and Company, 1979.
- [18] M. M. Halldórsson, J. Radhakrishnan. Greed is Good: Approximating Independent Sets in Sparse and Bounded-degree Graphs. *Algorithmica*, 18:143–163, 1997.
- [19] M. Hota, M. Pal, T. K. Pal. An Efficient Algorithm for Finding a Maximum Weight k -Independent Set on Trapezoid Graphs. *Computational Optimization and Applications*, 18(1):49–62, 2001.
- [20] S. Janson, T. Łuczak, A. Ruciński. *Random Graphs*. John Wiley and Sons, 2000.
- [21] M.C. Kong, Y. Zhao. On Computing Maximum k -Independent Sets. *Congressus Numerantium*, 95:47–60, 1993.
- [22] M.C. Kong, Y. Zhao. Computing k -Independent Sets for Regular Bipartite Graphs. *Congressus Numerantium*, 143:65–80, 2000.
- [23] S. Kutten, D. Peleg. Fast Distributed Construction of Small k -dominating Sets and Applications. *Journal of Algorithms*, 28(1):40–66, 1998.
- [24] M. Lewenstein, Z. Gotthilf. Tighter approximation on greedy for maximum induced matchings in regular graphs. In Proceedings of WAOA 2005: *the 3rd Workshop on Approximation and Online Algorithms*, volume 3879 of *Lecture Notes in Computer Science*. To appear.

- [25] J. Liu, H. Zhou. Maximum induced matchings in graphs. *Discrete Mathematics*, 170:277–281, 1997.
- [26] L. Lovász, M. D. Plummer. *Matching Theory*, volume 29 of *Annals of Discrete Mathematics*. North Holland, 1986.
- [27] B.D. McKay. Independent Sets in Regular Graphs of High Girth. *Ars Combinatoria*, 23A:179–185, 1987.
- [28] R. Motwani. Average-case analysis of algorithms for matchings and related problems. *Journal of the Association for Computing Machinery*, 41(6):1329–1356, November 1994.
- [29] T. Nierhoff. *The k -Center Problem and r -Independent Sets*, Doctoral Thesis, Humboldt-Universität zu Berlin, 1999.
- [30] A. Steger, M. Yu. On induced matchings. *Discrete Mathematics*, 120:291–295, 1993.
- [31] L. J. Stockmeyer, V. V. Vazirani. NP-Completeness of some generalizations of the maximum matching problem. *Information Processing Letters*, 15(1):14–19, August 1982.
- [32] N. C. Wormald. Differential equations for random processes and random graphs. *Annals of Applied Probability*, 5:1217–1235, 1995.
- [33] N. C. Wormald. The differential equation method for random graph processes and greedy algorithms. In M. Karoński and H. J. Prömel, editors, *Lectures on Approximation and Randomized Algorithms*, pages 73–155. PWN, Warsaw, 1999.
- [34] N. C. Wormald. Analysis of greedy algorithms on graphs with bounded degrees. *Discrete Mathematics*, 273:235–260, 2003.
- [35] M. Zito. Induced matchings in regular graphs and trees. In P. Widmayer, G. Neyer, and S. Eidenbenz, editors, *Graph Theoretic Concepts in Computer Science; 25th International Workshop, WG'99*, volume 1665 of *Lecture Notes in Computer Science*, pages 89–100. Springer-Verlag, 1999.
- [36] M. Zito. *Randomised Techniques in Combinatorial Algorithmics*. Doctoral Thesis, Department of Computer Science, University of Warwick, 1999.