Large k-Separated Matchings of Random Regular Graphs

Mihalis Beis[†] **William Duckworth**[*] **Michele Zito**[†]

[†]Department of Computer Science
University of Liverpool
Liverpool, L69 7ZF, UK

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

Abstract

A k-separated matching in a graph is a set of edges at distance at least k from one another (hence, for instance, a 1-separated matching is just a matching in the classical sense). We consider the problem of approximating the solution to the maximum k-separated matching problem in random r-regular graphs for each fixed integer k and each fixed $r \geq 3$. We prove both constructive lower bounds and combinatorial upper bounds on the size of the optimal solutions.

1 Introduction

A *regular graph 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 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 vertices u_i and v_j. For any positive integer k, a k*-(separated) matching* of a graph, is a set of edges, $\mathcal{M}$, with the additional constraint that the minimum distance between any two edges in $\mathcal{M}$ is at least k (the qualifier "separated" will normally be omitted in the remainder of this paper). Let $\nu_k(G)$ be the size of the largest k-matchings in G. The *maximum* k*-matching* (MkM) problem asks for a k-matching of size $\nu_k(G)$. For $k = 1$ this is the classical maximum matching problem. Stockmeyer and Vazirani (Stockmeyer & Vazirani 1982) introduced the generalisation 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 M2M 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 (Erdős 1988, Faudree, Gyárfas & Tuza 1989, Liu & Zhou 1997, Steger & Yu 1993). The separation constraint imposed on the matching edges when $k > 2$ is a distinctive feature of the MkM problem and the main motivation for our algorithmic investigation of such problems.

MkM is NP-hard (Stockmeyer & Vazirani 1982) for each $k \geq 2$ (polynomial time solvable (Edmonds 1965) for $k = 1$). Improved complexity results are known for M1M (Motwani 1994) on random instances. In particular it has been proven that simple greedy heuristics a.a.s. (asymptotically almost surely) produce sets of $\frac{n}{2} - o(n)$ independent edges (Aronson, Frieze & Pittel 1998) in dense random graphs and random regular graphs. A number of results are known on the approximability of an optimal 2-matching (Cameron 1989, Duckworth, Manlove & Zito 2000, Zito 1999a). Zito (1999) presented some simple results on the approximability of an optimal 2-matching in dense random graphs.

In this paper, we consider rather natural heuristics for approximating the solution to the MkM problem, for each positive integer k, and analyse their performance on random regular graphs. We also prove combinatorial upper bounds on $\nu_k(G)$ using a direct expectation argument. The algorithm we present for M2M was analysed deterministically by Duckworth, Manlove & Zito (2000) where it was shown to return a 2-matching of size at least $r(n-2)/2(2r-1)(r-1)$ 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. For the case $r = 3$, the cardinality of a largest 2-matching $\mathcal{M}$ of a random 3-regular graph a.a.s. satisfies $0.26645n \leq |\mathcal{M}| \leq 0.282069n$ (Duckworth, Wormald & Zito 2002) (unfortunately the optimistic $0.270413n$ lower bound claimed in the paper is not correct).

In the following section we present the model used for generating regular graphs u.a.r. (uniformly at random) and Section 3 gives the randomised algorithms. The analyses of our algorithms uses differential equations and a Theorem of Wormald (2001) which we restate in Section 4. The following Theorem encompasses the results of this paper. The proof concerning the lower bounds appears in Section 5 and in the final section of this paper we prove the upper bounds.

Theorem 1 *For each fixed positive integer* k *and fixed integer* $r \geq 3$ *there exist two positive real numbers* $\lambda_k = \lambda_k(r)$ *and* $\mu_k = \mu_k(r)$ *such that* $\lambda_k n \leq \nu_k(G) \leq \mu_k n$ *a.a.s. if* G *is an* r*-regular graph on* n *vertices selected u.a.r.*

The table below reports the values of λ_k and μ_k for the first few values of r and k. A weaker version of Theorem 1 was proven in Beis, Duckworth & Zito (2002) .

	M1M		M2M		M3M		M4M	
r	λ_1	μ_1	λ_2	μ_2	λ_3	μ_3	λ_4	μ_4
3	0.5	0.5	0.2664	0.2821	0.1264	0.1561	0.0579	0.0946
4	0.5	0.5	0.2295	0.25	0.0798	0.1076	0.0236	0.0501
5	0.5	0.5	0.2046	0.227	0.0559	0.0793	0.0117	0.0294
6	0.5	0.5	0.1861	0.2092	0.0417	0.0611	0.0067	0.0186
7	0.5	0.5	0.1715	0.1947	0.0326	0.0488	0.0042	0.0126
8	0.5	0.5	0.1596	0.1826	0.0263	0.0399	0.0028	0.0089
9	0.5	0.5	0.1496	0.1724	0.0216	0.0335	0.0019	0.0065
10	0.5	0.5	0.141	0.1634	0.0182	0.0285	0.0014	0.0049

2 Uniform Generation of Random Regular Graphs

Let $\mathcal{G}(n,r\text{-reg})$ denote the uniform probability space of r-regular graphs on n vertices. A well known construction that gives uniformly distributed elements of $\mathcal{G}(n,r\text{-reg})$ is the *configuration model* (see, for example, Chapter 9 in Janson, Łuczak & Rucínski (2000)). 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 pairing $F \in \Omega$ corresponds to an r-regular (multi)graph with vertex set $V = \{1, \ldots, n\}$ 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. A pairing $F \in \Omega^*$ corresponds to a simple r-regular graph G with vertex set $V = [n]$, that is a regular graph without loops or multiple edges. Since each simple graph corresponds to exactly $(r!)^n$ pairings , a regular graph can be chosen u.a.r. by choosing a pairing F u.a.r. and rejecting the result if it contains loops or multiple edges. Notice that the first point in a random pair may be selected using any rule whatsoever, as long as the second point in that random pair is chosen u.a.r. from all the remaining free (unpaired) points. This preserves the uniform distribution of the final pairing. Notation $G \in \mathcal{G}(n,r\text{-reg})$ will signify that G is selected according to the model described above.

The configuration model gives a basis for proving properties of such 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 $\mathcal{G}(n,r\text{-reg})$.

3 The Algorithms

In this section we describe the simple greedy heuristics used to construct a large k-matching. The algorithms are quite general and may be applied to any graph. The analyses presented in Section 5 give lower bounds on the size of the resulting k-matching if the input graph is a random regular graph.

Dense Matchings

We now describe the algorithm that will be used to find a large k-matching in a random r-regular graph, when[1] $k \leq 2$. Let $\Gamma(u) = \{v \in G : \{u, v\} \in E\}$ be the *neighbourhood of vertex u*.

Algorithm DegreeGreedy(G, k)
Input: a graph $G = (V, E)$ on n vertices.
  $\mathcal{M} \leftarrow \emptyset$;
  while $E \neq \emptyset$
    pick a vertex u of minimum positive degree in $V(G)$;
    pick a vertex v of minimum positive degree in $\Gamma(u)$;
    $\mathcal{M} \leftarrow \mathcal{M} \cup \{\{u, v\}\}$;
    shrink$(G, \{u, v\}, k)$;

For each iteration of the algorithm, procedure *shrink* updates G by removing all edges incident with vertices at distance at most $k - 1$ from $\{u, v\}$.

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

Sparse Matchings

Any obvious adaptation of the algorithm DegreeGreedy to the case $k > 2$ fails. The DegreeGreedy process, which repeatedly picks sparsely connected edges $\{u, v\}$ and removes their neighbourhood at distance at most $k - 1$, has no permanent record of the original neighbourhood structure of each vertex. Hence, an edge chosen to be added to $\mathcal{M}$ may cause the matching not to be k-separated. For $k > 2$ we therefore resort to different algorithms. Such algorithms are based on the idea of repeatedly removing induced copies of a particular type of tree from the given graph G. 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 $T_k(r)$ is a rooted tree whose root u has a child v which is the root of a copy of $t_{\lfloor k/2 \rfloor}(r-1)$ and $r - 1$ other children $v_2, \ldots, v_r$ which are roots of copies of $t_{\lfloor k/2 \rfloor - 1}(r-1)$. In other words, a copy of $T_k(r)$ consists of two complete $(r-1)$-ary trees of depth $\lfloor \frac{k}{2} \rfloor$ whose roots are connected by an edge $e_T = \{u, v\}$. The matching algorithms used for $k > 2$ will repeatedly try to find induced copies of $T_k(r)$ in G, add e_T to $\mathcal{M}$ and remove all edges in $T_k(r)$ from G.

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 $T_k(r)$, and they are not all equivalent in terms of the cardinality of the matching returned. It turns out that the best alternative is to start exploring a possible copy of $T_k(r)$ from one of its leaves and, furthermore to take as a candidate leaf a vertex of minimum degree in G.

It is therefore convenient to talk of the vertices in $T_k(r)$ 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 $0 < l < 2\lfloor \frac{k}{2} \rfloor + 1$) 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 $2\lfloor \frac{k}{2} \rfloor + 1$ is composed of $(r-1)^{\lfloor \frac{k}{2} \rfloor}$ leaves only. The k-matching algorithm may be described as follows:

Algorithm Sparse(G, k)
Input: an r-regular graph $G = (V, E)$ on n vertices.
  $\mathcal{M} \leftarrow \emptyset$;
  while *possible*(G)
(*)   pick a vertex w of minimum positive degree in $V(G)$;
    uncover a copy of $T_k(r)$ by marking candidate edges
      one level at a time;
    if (a copy of $T_k(r)$ has been found)
      $\mathcal{M} \leftarrow \mathcal{M} \cup e_T$;
(**)   *shrink*(G, k);

The implementation of steps (*) and (**) depends on k. If k is even w will be picked, if possible, as the same vertex from where the search for $T_k(r)$ started in the previous iteration of the main while loop, otherwise a random selection among the vertices of minimum positive degree will suffice.

The implementation of *shrink* also depends on the parity of k. If k is odd then distinct induced copies of $T_k(r)$ must also be vertex disjoint. Therefore all edges incident to the leaves of $T_k(r)$ must be removed as well.

4 Analysing Algorithms using Differential Equations

In order to approximate the expected size of the k-matching returned by our algorithms, we use a result of Wormald (2001), the setting of which requires the following general definitions (Wormald 2001).

Denote by G_0 the initial r-regular graph and by G_t for $t \geq 0$ the subgraph of the input graph still to be dealt with at step t of the execution of the algorithm. The execution of any of our algorithms consists of a sequence of operations op_t, $t \geq 0$ with each operation being one of Op_i, $i = 1, \ldots, r$, where Op_i consists of selecting a vertex v of degree i in G_t u.a.r., and then applying some specified set of tasks, to obtain G_{t+1}. A subset $\mathcal{M}$ of $E(G)$ is selected during the operations, with $\mathcal{M}_0 = \emptyset$ initially, and $\mathcal{M} = \mathcal{M}_t$ for the graph G_t. For $0 \leq i \leq r$, let $Y_i = Y_i(t)$ denote the number of vertices of degree i in G_t. Also let Y_{r+1} denote the cardinality of the set $\mathcal{M}_t$.

Assume that the expected change in Y_i, in going from G_t to G_{t+1}, conditional upon G_t and op_t, is determined approximately, depending only upon t, op_t, and $Y_1(t), \ldots, Y_{r+1}(t)$. In some sense, this is a measure of the rate of change of Y_i. We express the assumption by asserting that for some fixed functions $f_{i,q}(x, \mathbf{y}) = f_{i,q}(x, y_1, \ldots, y_{r+1})$,

$$\mathbf{E}\Big(Y_i(t+1) - Y_i(t) \mid G_t \wedge \{\mathrm{op}_t = \mathrm{Op}_q\}\Big) = f_{i,q}\left(\tfrac{t}{n}, \tfrac{Y_1}{n}, \ldots, \tfrac{Y_{r+1}}{n}\right) + o(1) \tag{1}$$

for $i = 1, \ldots, r+1$, $q = 1, \ldots, r$ such that $Y_q(t) > 0$. The convergence in $o(1)$ is uniform over all appropriate choices of t and G_t as functions of n with certain restrictions on G_t which will be specified. Uniformity over q and i then follows, since there are finitely many possibilities for these two variables.

Since the initial graph is an r-regular graph on n vertices, the first operation must apply $\mathrm{op}_0 = \mathrm{Op}_r$. This typically produces some vertices of degree less than r, so the next operation is determined by their minimum degree. Both Op_r and Op_{r-1} typically produce vertices of degree $r-1$ but none of smaller (positive) degrees when Y_{r-1} is small (say $o(n)$), so the second operation normally involves Op_{r-1}, as does the next, and this remains so until a vertex of smaller degree, say $r-2$, is produced. This causes a temporary hiccup, with an Op_{r-2}, followed by more operations of Op_{r-1}. When vertices of degree $r-1$ become plentiful, vertices of smaller degree are more commonly created, and the hiccups occur more often. Suppose that at some time t in the process, an Op_{r-1} creates, in expectation, α vertices of degree $r-2$, and an Op_{r-2} decreases the number of vertices of degree $r-2$, in expectation, by τ. Then we expect each Op_{r-1} to be followed by (on average) α/τ operations of Op_{r-2}. At some stage τ may fall below 0, at which point the vertices of degree $r-2$ begin to build up and do not decrease under repeated applications of Op_{r-2}. Then vertices of degree $r-2$ take over the role of vertices of degree $r-1$, and we say *informally* that the first phase of the process has finished and the second has begun. The process may continue through further phases; typically, the jth phase begins with an increasing abundance of vertices of degree $r-j$. Note that by the assumptions above, the asymptotic values of α and τ in the first phase are the cases $j = 1$ of the general definitions

$$\begin{aligned} \alpha_j(x, \mathbf{y}) &= f_{r-j-1, r-j}(x, \mathbf{y}), \\ \tau_j(x, \mathbf{y}) &= -f_{r-j-1, r-j-1}(x, \mathbf{y}), \end{aligned} \tag{2}$$

where

$$x = \frac{t}{n}, \qquad \mathbf{y}(x) = \frac{\mathbf{Y}(t)}{n}. \tag{3}$$

Since each Op_{r-j} is followed (on average) by α_j/τ_j operations of Op_{r-j-1}, we expect the proportion of operations involving an operation of the former type to be $1/(1 + \alpha_j/\tau_j) = \tau_j/(\tau_j + \alpha_j)$, and of the latter type to be $\alpha_j/(\tau_j + \alpha_j)$. This *suggests* that, if y_i as prescribed in (3) were a differentiable function of a real variable, its derivative would satisfy

$$\frac{\mathrm{d}y_i}{\mathrm{d}x} = F(x, \mathbf{y}, i, j) \tag{4}$$

where $F(x, \mathbf{y}, i, j)$ is equal to

$$\frac{\tau_j}{\tau_j + \alpha_j} f_{i, r-j}(x, \mathbf{y}) + \frac{\alpha_j}{\tau_j + \alpha_j} f_{i, r-j-1}(x, \mathbf{y}) \tag{5}$$

if $j \leq r-2$, and it is just

$$f_{i,1}(x, \mathbf{y}) \tag{6}$$

for $j = r-1$. Our assumptions will ensure that the phases proceed in an orderly fashion, and that the last possible phase is $j = r-1$, in which all operations are Op_1.

We will work with the parameters of $f_{i,\ell}$ in the domain

$$\begin{aligned} \mathcal{D}_\epsilon = \{(x, \mathbf{y}) :\ & 0 \leq x \leq r, \\ & 0 \leq y_i \leq r \\ \text{for}\quad & 1 \leq i \leq r+1,\ y_r \geq \epsilon\} \end{aligned} \tag{7}$$

for some pre-chosen value of $\epsilon > 0$. The behaviour of the process will be described in terms of the function $\tilde{\mathbf{y}} = \tilde{\mathbf{y}}(x) = (\tilde{y}_1(x), \ldots, \tilde{y}_{r+1}(x))$ defined as follows, with reference to an initial value $x = x_0 = t_0/n$ of interest:

> $\tilde{y}_i(x_0) = Y_i(t_0)/n$, $i = 1, \ldots, r+1$, and inductively for $j \geq 1$, $\tilde{\mathbf{y}}$ is the solution of (4) with initial conditions $\mathbf{y}(x_{j-1}) = \tilde{\mathbf{y}}(x_{j-1})$, extending to all $x \in [x_{j-1}, x_j]$, where x_j is defined as the infimum of those $x > x_{j-1}$ for which at least one of the following holds: $\tau_j \leq 0$ and $j < r-1$; $\tau_j + \alpha_j \leq \epsilon$ and $j < r-1$; $\tilde{y}_{r-j} \leq 0$; or the solution is outside $\mathcal{D}_\epsilon$ or ceases to exist. (8)

The interval $[x_{j-1}, x_j]$ represents phase j, and the termination condition $\tilde{y}_{r-j} = 0$ is necessary to ensure that the process does not revert to the conditions of phase $j-1$. Typically it will eventuate that $\tilde{y}_{r-j}(x_{j-1}) = 0$ but $\tilde{y}_{r-j}(x) > 0$ for x greater than, but close to, x_{j-1}, which permits phase j to endure for a non-empty interval $[x_{j-1}, x_j]$, provided τ_j stays positive on such an interval. We require that this inductive definition of $\tilde{\mathbf{y}}$ continues for phases $j = 1, 2, \ldots, m$, where

> m denotes the smallest j for which either $j = r-1$, or any of the termination conditions for phase j in (8) hold at x_j apart from $x_j = \inf\{x \geq x_{j-1} : \tau_j \leq 0\}$. (9)

We will impose conditions to ensure that the intervals in the definition of $\tilde{\mathbf{y}}$ representing phases $1, 2, \ldots, m$ are nonempty. These conditions are firstly

$$\begin{aligned} \tau_j &> 0 \quad \text{and} \\ \tau_j + \alpha_j &> \epsilon \quad \text{at } (x_{j-1}, \tilde{\mathbf{y}}(x_{j-1})) \\ \text{for} &\quad 1 \leq j \leq \min\{r-2, m\}. \end{aligned} \tag{10}$$

Also

$$
\begin{array}{ll}
f_{r-1,r-1} > 0 & \text{at} \quad (x_0, \tilde{\mathbf{y}}(x_0)), \\
f'_{r-j,r-j}\tau_j + f_{r-j,r-j-1}f'_{r-j-1,r-j} > 0 & \\
\quad \text{at} \quad (x_{j-1}, \tilde{\mathbf{y}}(x_{j-1}))^+ & \\
\quad \text{for} \quad 1 < j \leq \min\{r-2, m\}, & \\
f'_{r-j,r-j} > 0 \quad \text{at} \quad (x_{j-1}, \tilde{\mathbf{y}}(x_{j-1}))^- & \\
\quad \text{for} \quad 1 < j \leq m, & \\
f'_{1,1} > 0 \quad \text{at} \quad (x_{r-2}, \tilde{\mathbf{y}}(x_{r-2}))^+ & \\
\quad \text{if } m = r-1, &
\end{array} \tag{11}
$$

with f' denoting $\frac{\mathrm{d}f(x,\tilde{\mathbf{y}}(x))}{\mathrm{d}x}$ and $(x, \tilde{\mathbf{y}}(x))^+$ and $(x, \tilde{\mathbf{y}}(x))^-$ referring to the right-hand and left-hand limits as functions of x.

We now restate (Wormald 2001, Theorem 1) which we will use in the following section to analyse the performance of our algorithms.

Theorem 2 ((Wormald 2001)) *Let $r \geq 3$, for $1 \leq i \leq r$ let $Y_i(t)$ denote the number of vertices of degree i in G_t, and let $Y_{r+1}(t)$ denote $|\mathcal{M}_t|$. Assume that for some fixed $\epsilon > 0$ the operations Op_q satisfy (1) for some fixed functions $f_{i,q}(x, y_1(x), \ldots, y_{r+1}(x))$ and for $i = 1, \ldots, r+1$, $q = 1, \ldots, r$, with the convergence in $o(1)$ uniform over all t and G_t for which $Y_q(t) > 0$ and $Y_r(t) > \epsilon n$. 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 $\mathcal{M}$ (i.e. $|\mathcal{M}_{t+1}| - |\mathcal{M}_t|$), during any one operation;

(ii) the functions $f_{i,q}$ are rational functions of $x, y_1, \ldots, y_{r+1}$ with no pole in $\mathcal{D}_\epsilon$ defined in (7);

(iii) there exist positive constants C_1, C_2 and C_3 such that for $1 \leq i < r$, everywhere on $\mathcal{D}_\epsilon$, $f_{i,q} \geq C_1 y_{i+1} - C_2 y_i$ when $q \neq i$, and $f_{i,q} \leq C_3 y_{i+1}$ for all q.

Define $\tilde{\mathbf{y}}$ as in (8), set $x_0 = 0$, define m as in (9), and assume that (10) and (11) both hold. Then there is a randomised algorithm for which a.a.s. there exists t such that $|\mathcal{M}_t| = n\tilde{y}_{r+1}(x_m) + o(n)$ and $Y_i(t) = n\tilde{y}_i(x_m) + o(n)$ for $1 \leq i \leq r$. Also $\tilde{y}_i(x) \equiv 0$ for $x_{j-1} \leq x \leq x_j$, $1 \leq i \leq r - j - 1$ $(1 \leq j \leq m)$.

5 Algorithm Analyses

5.1 Analysis for $k \leq 2$

The results proven in this section are identical to those reported for $k \leq 2$ in (Beis, Duckworth & Zito 2002). 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 (Beis, Duckworth & Zito 2002).

Consider the algorithm DegreeGreedy for finding a large k-matching ($k \leq 2$) as described in Section 3. Here, in the specification of Op_q operation (which selects a random vertex u of degree q) the set of randomised tasks consists of choosing a vertex v and removing all edges incident with vertices at distance at most $k-1$ from $\{u, v\}$.

We may verify the hypotheses of Theorem 2. First we will show that (1) holds when $Y_r(t) > \epsilon n$ (for any $\epsilon > 0$).

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

Let $X = \sum_{i=1}^{r} iY_i$. The probability of creating a vertex of degree $i-1$ when removing an edge chosen at random from those incident to some given vertex in the graph is asymptotically:

$$P_i = \frac{iY_i}{X}$$

Denote by S_a^b the sum of all P_i's for $a \leq i \leq b$. The expected change in Y_i due to removing an edge from a vertex of unknown degree can be approximated by $-Q_i(0)$ where:

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

Using the same reasoning the expected change in Y_i due to the removal of a random edge incident to a given vertex in the graph and the removal of any other edge incident to the other end-point of the initial edge is asymptotically $-Q_i(1)$ where:

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

We calculate the expected change in Y_i when performing an Op_q operation (i.e. selecting a vertex u of degree q) 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 of a vertex in $\Gamma(u)$ is x, is $\chi_x + o(1)$ where

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

The expected change in Y_i, conditioned to the minimum degree in $\Gamma(u)$ being x, 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 $\phi_{i,x} + o(1)$ where

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

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 $\beta_{x,d} + o(1)$ where

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

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 $\psi_{i,x,d} + o(1)$ where

$$\begin{aligned}\psi_{i,x,d} = (d-1)(&-[i = x] + \\ &+[k = 1 \wedge i = x-1] - [k = 2](x-1)Q_i(0))\end{aligned}$$

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 $\varepsilon_{x,d,m} + o(1)$ where

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

with the convention that the expected value is zero if $x = r$.

The expected change in Y_i due to the removal of any edge incident with a vertex of degree m in $\Gamma(u)$, is $\gamma_{i,x,d,m} + o(1)$ where

$$\begin{aligned}\gamma_{i,x,d,m} = &-[i = m] \\ &+[k = 1 \wedge i = m-1] - [k = 2](m-1)Q_i(0)\end{aligned}$$

Finally, the asymptotic expression for $f_{i,q}$ can be written as

$$-[i=q]+\sum_{x=q}^{r}\chi_x\left(\phi_{i,x}+\sum_{d=1}^{q}\beta_{x,d}\left(\psi_{i,x,d}+\sum_{m=x+1}^{r}\varepsilon_{x,d,m}\gamma_{i,x,d,m}\right)\right)1\le i\le r \quad (12)$$

and $f_{r+1,q}=1$ since an edge is added to $\mathcal{M}$ following each Op_q.

Hypothesis (i) of the theorem is immediate since in any operation only the edges involving the selected vertex and its neighbours are deleted, and a bounded number of edges are added to $\mathcal{M}$. The functions $f_{i,q}$ satisfy (ii) because from (12) their (possible) singularities satisfy $s=0$, which lies outside $\mathcal{D}_\epsilon$ since in $\mathcal{D}_\epsilon$, $s\ge y_r\ge\epsilon$. Hypothesis (iii) follows from (12) again using $s\ge y_r\ge\epsilon$ and the boundedness of the functions y_i (which follows from the boundedness of $\mathcal{D}_\epsilon$). Thus, defining $\tilde{\mathbf{y}}$ as in (8) with $t_0=0$, $Y_r(0)=n$ and $Y_i(0)=0$ for $i\ne r$, we may solve (4) numerically to find m, verifying (10) and (11) at the appropriate points of the computation. It turns out that these hold for each r which was treated numerically and that in each case $m=r-2$, 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 2, this is the asymptotic value of the size of $\mathcal{M}$ at the end of some randomised algorithm. So the conclusion is that a random r-regular graph a.a.s. has a k-matching of size at least $n\tilde{y}_{r+1}(x_m)+o(n)$.

Note that (by Theorem 2) $\tilde{y}_i(x)\equiv 0$ in phase j for $1\le i\le r-j-1$, and by the nature of the differential equation, $\tilde{y}_i(x)$ will be strictly positive for $i>r-j$. So by (8) and (9), the end of the process (for ϵ arbitrarily small) occurs in phase $r-2$ when either $\tau_j+\alpha_j\le\epsilon$ or $\tilde{y}_2$ becomes 0. Numerically, we find it is the latter. This is numerically more stable as a check for the end of the process than checking when $\tilde{y}_r$ reaches 0, since the derivative of the latter is very small.

5.2 Analysis for $k>2$

The machinery used to analyse the algorithm for $k\le 2$ is equally well suited for this case. To complete the analysis we only need to define the randomised tasks specified by an Op_q operation and the asymptotic expressions for $f_{i,q}$. Details depend again on the parity of k and are given, separately, in the forthcoming sections.

5.2.1 Details for even k.

An Op_q operation consists of the selection of a random vertex w of degree q followed by an attempt of finding q copies of $T_k(r)$ around w.

The asymptotic expression for $f_{i,r}$ is

$$-[i=q]-q\Bigg(Q_i(0)+\sum_{l=0}^{k-1}\prod_{m=0}^{l}\mathcal{P}_m\left((r-1)^{\lfloor l/2\rfloor}[i=r-1]+(r-1)^{\lfloor l/2\rfloor+1}Q_i(0)\right)\Bigg)\quad\text{for}\quad 1\le i\le r \quad (13)$$

where $\mathcal{P}_m$ represents the probability of succeeding at level m. 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 $(P_r)^{(r-1)^{\lfloor m/2\rfloor}}$ when m is even, and it is $1-(1-\mathcal{P}_{m-1})^{r-1}$ otherwise. If success does occur at level l (i.e. the right event happens at level $l+1$) then the previously accounted for contribution to $f_{r-1,q}$ given by the removal of a single edge incident to each of the $(r-1)^{\lfloor l/2\rfloor}$ 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 add an edge to the matching during an Op_q operation.

5.2.2 Details for odd k.

Here, an Op_q operation consists of the selection of a random vertex w of degree q followed by an attempt of revealing a $T_k(r)$ structure around w. The analysis for this case is complicated by the requirement that distinct copies of $T_k(r)$ must be vertex disjoint. The expected change in Y_i can be computed, as in the case $k\le 2$, by conditioning on the degree distribution in $\Gamma(w)$ but major differences arise. First of all we must condition on the maximum degree in $\Gamma(w)$ 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 $T_k(r)$ (otherwise the current trial has no hope of finding a copy of $T_k(r)$). Finally the major complication in the asymptotic expression for $f_{i,q}$ comes from the need to be able to delete all edges incident with the leaves of $T_k(r)$ and these leaves (in the graph) can have arbitrary degree.

Since when performing an Op_q operation all vertices have degree at least q, the probability that the maximum degree among the vertices whose degree is affected by the removal of $(r-1)^c$ edges be x, with $q\le x\le r$ is $\chi_x^c+o(1)$, where

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

The probability of having exactly d vertices of such maximum degree in the experiment outlined above is $\beta_{x,d}^c+o(1)$ where $\beta_{x,d}^c$ is equal to

$$\frac{\binom{(r-1)^c}{d}(P_x)^d(S_q^{x-1})^{(r-1)^c-d}}{\chi_x^c}$$

but $\beta_{q,(r-1)^c}^c=1$ and $\beta_{x,d}^c=0$ if $x=q$ and $d\ne(r-1)^c$ or $\chi_x^c=0$.

The expected number of vertices of degree m, with $q\le m<x$, given that the maximum degree of the vertices affected by the $(r-1)^c$ edge removals was x and that there are d vertices of such degree, is $\varepsilon_{x,d,m}^c+o(1)$ where

$$\varepsilon_{x,d,m}^c=\begin{cases}0 & \text{if } S_q^{x-1}=0\\ ((r-1)^c-d)\frac{P_m}{S_q^{x-1}} & \text{otherwise}\end{cases}$$

The expected change in Y_i due to the removal of all remaining edges out of a vertex of (initially) degree m, given that it has d siblings of maximum degree x is $\mathrm{chg}_{i,x,d,m}+o(1)$ where

$$\mathrm{chg}_{i,x,m}=-[i=m]+[x\ne r\wedge i=m-1]-[x=r](m-1)Q_i(0)$$

Finally, the asymptotic expression for $f_{i,q}$ can be written as

$$f_{i,q}= \\ = -[i=q]-qQ_i(0)+$$

$$(1-(1-P_r)^q)\Big(-[i=r-1]+$$
$$\sum_{x=q}^{r}\chi_x^1\Big((-[i=x]+[x\neq r\wedge i=x-1]+[x=r]\Xi_1^{\lfloor\frac{k}{2}\rfloor})+$$
$$\sum_{d=1}^{d-1}\beta_{x,d}^1\big((r-1)\mathrm{chg}_{i,x,x}+\sum_{m=q}^{x-1}\varepsilon_{x,d,m}^1\mathrm{chg}_{i,x,m})\Big)\Big)$$
$$\text{for}\qquad 1\le i\le r$$

and

$$f_{r+1,q}=(1-(1-P_r)^q)\mathcal{P}_0\prod_{l=1}^{\lfloor\frac{k}{2}\rfloor-1}P_r^{(r-1)^l}\mathcal{P}_l$$

$\Xi_1^{\lfloor\frac{k}{2}\rfloor-1}$ models the behaviour of the algorithm from level two onwards. For example for $k=3$ it is $\Xi_1^0=-(r-1)Q_i(1)$. 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 $d\ge(r-1)^a+(r-2)((r-1)^a-1)$ the success is certain. In any other case we may have either a failure or a success.
In case of a failure the behaviour of the algorithm can be described by

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

Let $(r-1)^a\le d<(r-1)^a+(r-2)((r-1)^a-1)$, 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 either it starts the above procedure all over again or it has reached the final level. The former case (where it starts again) can be described by the following recursion.

$$\Xi_a^b = -(r-1)^aQ_i(0)+P_r^{(r-1)^a}\Bigg(-[i=r-1](r-1)^a+$$
$$\sum_{x=q}^{r-1}\chi_x^{a+1}\Big(\sum_{d=1}^{(r-1)^{a+1}}\beta_{x,d}^{a+1}\big(d(-[i=x]+[i=x-1])+$$
$$\sum_{m=q}^{x-1}\varepsilon_{x,d,m}^{a+1}(-[i=m]+[i=m-1])\big)\Big)+$$
$$\chi_r^{a+1}\Big(\sum_{d=1}^{(r-1)^{a+1}}\beta_{r,d}^{a+1}([d<(r-1)^a]A_a+$$
$$[d\ge(r-1)^a+(r-2)((r-1)^a-1)](\Xi_{a+1}^b+\xi_{i,d}^a)+$$
$$[d\ge(r-1)^a\wedge d<(r-1)^a+(r-2)((r-1)^a-1)]B_a)\Big)\Bigg)$$

with

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

The latter case (where the algorithm has reached the final level) can be described by the following base case

$$\Xi_b^b=-(r-1)^bQ_i(1)$$

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

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

6 Upper Bounds

Let $G\in\mathcal{G}(n,r\text{-reg})$ be a graph generated using the configuration model given in Section 2. Let Q_k be the event "G contains at least one k-matching of size y". We will show that there exists a positive real number μ_k such that when $\mu>\mu_k$ and $y=\mu n$ then $\Pr[Q_k]=o(1)$, thus proving the upper bound in Theorem 1. Let Q'_k be the corresponding event defined in $F\in\Omega$ (that is, on pairings that correspond to n-vertex r-regular multigraphs). Since any event holding a.a.s. for a random pairing also holds a.a.s. for a random graph in $\mathcal{G}(n,r\text{-reg})$ we can estimate $\Pr[Q_k]$ by performing all our calculations on random pairings.

Let $X_k=X_k(F,r,n,y)$ be the number of k-matchings in a random pairing. We calculate an asymptotic expression for $\mathrm{E}(X_k)$ and show that when $y=\mu_kn$, then $\mathrm{E}(X_k)=o(1)$, thus proving the upper bound in Theorem 1 through Markov's inequality. Let $X_{\mathcal{M}}$ be a random indicator which equals one if $\mathcal{M}$ is a k-matching of size y in a random pairing and zero otherwise. Then $\mathrm{E}(X_k)=\sum_{\mathcal{M}}\mathrm{E}(X_{\mathcal{M}})=\binom{n}{2y}\mathrm{E}(X_{\mathcal{M}})$. Notice that $\mathrm{E}(X_{\mathcal{M}})$ is the probability that a set of $2y$ vertices forms a k-matching which is equal to[2] $R_k/(rn-1)!!$. Thus

$$\mathrm{E}(X_k)=\binom{n}{2y}\frac{R_k}{(rn-1)!!}$$

with R_k being the number of configurations in which $\mathcal{M}$ is a k-matching. Such configurations have a very special structure. It is well known (Janson, Łuczak & Rucínski 2000) that a.a.s. random regular graphs contain very few short cycles. This implies that R_k can be computed by counting the number of ways in which y copies of (pairings corresponding to) $T_k(r)$ can be "embedded" in a pairing. There are $(2y-1)!!r^{2y}$ ways of forming the matching edges. The remaining $2y(r-1)$ points in $\mathcal{M}$ (the "matching urns") can be paired in $r^{2y(r-1)}(n-2y)_{2y(r-1)}$ ways. We select

[2] The semifactorial is defined to be $n!!=n(n-2)\cdots3\cdot1=(2m)!/m!2^m$ when $n=2m-1$ is odd

$2y(r-1)^2$ more urns to pair the points at distance one from the matching edges (in the same way as above) and we repeat this until we have paired the points at distance $\lfloor k/2 \rfloor - 1$ (the final step involving the selection of the leaves of $T_k(r)$). There is a difference in the number of pairings, between k-odd and k-even, when we select the leaves of $T_k(r)$ (corresponding to urns at distance $\lfloor k/2 \rfloor$ from the matching edges) since only when k is odd the leaves must be distinct. Finally, if k is odd

$$R_k = r^{2y}(2y-1)!! \prod_{i=1}^{\lfloor \frac{k}{2} \rfloor} \left(r^{2y(r-1)^i} \left(n - 2y \sum_{j=0}^{i-1} (r-1)^j\right)_{2y(r-1)^i} \right) (\Upsilon - 1)!!$$

whereas if k is even

$$R_k = R_{k-1} \left(rn - r2y \sum_{j=0}^{\lfloor \frac{k}{2} \rfloor - 1} (r-1)^j\right)_{2y(r-1)^{\lfloor \frac{k}{2} \rfloor}}$$

with

$$\Upsilon = nr - \left(2y + 4y \sum_{i=1}^{\lfloor k/2 \rfloor} (r-1)^i\right)$$

Setting $y = \mu n$ and using the standard Stirling's approximation for the factorials the expressions for the expectations $\mathrm{E}(X_k)$ have the form $f(r, \mu)^n$. Therefore for each r there is $\mu_k(r) > 0$ such that $f(r, \mu) < 1$ for $\mu > \mu_k(r)$.

Maximal vs. non-maximal k-matchings.

It should be remarked that slightly smaller values of $\mu_k(r)$ than those reported in the table in Section 1 can be numerically computed counting *maximal* k-matchings (a stronger 0.28206915 value for $\mu_2(3)$ is reported in (Duckworth, Wormald & Zito 2002)). However we preferred to keep the simpler exposition presented above as the magnitude of the improvements (less than 10^{-5} for each $r \geq 3$ and $k = 2$ and even smaller for larger k) makes the more complicated analysis rather un-appealing.

References

Aronson, J., Frieze, A. & Pittel, B. G. (1998), Maximum matchings in sparse random graphs: Karp-Sipser revisited. *Random Structures and Algorithms*, 12:111–178.

Beis, M., Duckworth, W. & Zito, M. (2002), 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.

Cameron, K. (1989), Induced matchings. *Discrete Applied Mathematics*, 24(1-3):97–102.

Cash, J. R. & Karp, A. H. (1990), A variable order runge-kutta method for initial value problems with rapidly varying right-hand sides. *ACM Transactions on Mathematical Software*, 16(3):201–222.

Duckworth, W., Manlove, D. & Zito, M. (2000), On the approximability of the maximum induced matching problem. Technical Report TR-2000-56, Department of Computing Science, The University of Glasgow.

(2002), Duckworth, W., Wormald, N. C. & Zito, M. Maximum induced matchings of random cubic graphs. *Journal of Computational and Applied Mathematics*, 142(1):39–50.

Edmonds, J. (1965) Paths, trees and flowers. *Canadian Journal of Mathematics*, 15:449–467.

Erdős, P. (1988), Problems and results in combinatorial analysis and graph theory. *Discrete Mathematics*, 72:81–92.

Faudree, R. J., Gyárfas, A., Schelp, R. H. & Tuza, Z. (1989), Induced matchings in bipartite graphs. *Discrete Mathematics*, 78(1-2):83–87.

Gear, C. W. (1971), *Numerical Initial Value Problems in Ordinary Differential Equations.* Prentice-Hall.

Janson, S., Łuczak, T. & Rucínski, A. (2000), *Random Graphs.* John Wiley and Sons.

Liu, J. & Zhou, H. (1997), Maximum induced matchings in graphs. *Discrete Mathematics*, 170:277–281.

Lovász, L. & Plummer, M. D. (1986), *Matching Theory*, volume 29 of *Annals of Discrete Mathematics.* North Holland.

Motwani, R. (1994), Average-case analysis of algorithms for matchings and related problems. *Journal of the Association for Computing Machinery*, 41(6):1329–1356.

Steger, A. & Yu, M. (1993), On induced matchings. *Discrete Mathematics*, 120:291–295.

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

Wormald, N. C. (2001), Analysing Greedy Algorithms on Regular Graphs. In J. Nesetril, M. Noy and O. Serra (editors) *Electronic Notes in Discrete Mathematics*, **10**. Elsevier Science Publishers.

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

Wormald, N. C. (1999), 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.

Zito, M. (1999a), 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.

Zito, M. (1999b), *Randomised Techniques in Combinatorial Algorithmics.* PhD thesis, Department of Computer Science, University of Warwick.