Maximum Induced Matchings of Random Cubic Graphs*

W. Duckworth[1†], N.C. Wormald[1‡] and M. Zito[2§]

[1] Department of Mathematics & Statistics, University of Melbourne, Australia

[2] Department of Computer Science, University of Liverpool, UK

Abstract

We present a heuristic for finding a large induced matching of cubic graphs. We analyse the performance of this heuristic, which is a random greedy algorithm, on random cubic graphs using differential equations and obtain a lower bound on the expected size of the induced matching, $\mathcal{M}$, returned by the algorithm. A corresponding upper bound is derived by means of a direct expectation argument. We prove that $\mathcal{M}$ asymptotically almost surely satisfies $0.270413n \leq |\mathcal{M}| \leq 0.282069n$.

1 Introduction

An *induced matching* of a graph, $G{=}(V, E)$, is a set of vertex disjoint edges, $\mathcal{M} \subseteq E$, with the additional constraint that no two edges of $\mathcal{M}$ have their endpoints connected by an edge of $E \setminus \mathcal{M}$. We are interested in finding induced matchings of large cardinality.

Stockmeyer and Vazirani [12] introduced the problem of finding a maximum induced matching of a graph, motivating it 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). This in turn stimulated much interest in other areas of theoretical computer science and discrete mathematics as finding a maximum induced 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) [6, 7, 10, 11]).

The problem of deciding whether, for a given integer k, a given graph has an induced matching of size at least k is NP-complete [12], even for bipartite graphs of maximum degree 4. It has been shown [4, 15], that the optimisation version of the same problem is APX-complete, even when restricted to $3k$-regular or $4k$-regular graphs for any integer $k \geq 1$. Maximum induced matching is polynomial-time solvable for chordal graphs [2] and circular arc graphs [8].

*An earlier version of this paper appeared in the proceedings of "COCOON 2000"
†Currently in the Department of Computing, Macquarie University, Australia
‡Supported by the Australian Research Council
§Supported by EPSRC grant GR/L/77089

Recently, Golumbic and Lewenstein [9] have constructed polynomial-time algorithms for maximum induced matching in trapezoid graphs, interval-dimension graphs and co-comparability graphs and have given a linear-time algorithm for maximum induced matching in interval graphs.

In this paper, we present a heuristic for finding a large induced matching of cubic graphs. We analyse the performance of this heuristic, which is a random greedy algorithm, on random cubic graphs using differential equations and obtain a lower bound on the expected size of the induced matching, $\mathcal{M}$, returned by the algorithm. A corresponding upper bound is derived by means of a direct expectation argument. We prove that $\mathcal{M}$ asymptotically almost surely satisfies $0.270413n \le |\mathcal{M}| \le 0.282069n$.

Little is known on the complexity of this problem under the additional assumption that the input graphs occur with a given probability distribution. Zito [16] presented some simple results on dense random graphs.

The algorithm we present was analysed deterministically in [4] where it was shown that, given an n-vertex connected cubic graph, the algorithm returns an induced matching of size at least $3n/20 + O(1)$ and there exist infinitely many cubic graphs for which the algorithm only achieves this bound.

Throughout this paper we use the notation $\mathbf{P}$ (probability), $\mathbf{E}$ (expectation), u.a.r. (uniformly at random) and a.a.s. (asymptotically almost surely) (see, for example, [1] for these and other random graph theory definitions). When discussing any cubic graph on n vertices, we assume n to be even to avoid parity problems.

In the following section we introduce the model used for generating cubic graphs u.a.r. and in Section 3 we describe the notion of analysing the performance of algorithms on random graphs using a system of differential equations. Section 4 gives the randomised algorithm and Section 5 gives its analysis showing the a.a. sure lower bound. In Section 6 we give a direct expectation argument showing the a.a. sure upper bound.

2 Generating Cubic Graphs u.a.r.

The model used to generate a cubic graph u.a.r. (see, for example, Bollobás [1]) can be summarised as follows. For an n vertex graph

- take $3n$ points in n buckets labelled $1 \ldots n$ with three points in each bucket and
- choose u.a.r. a pairing of the $3n$ points.

If no pair contains two points from the same bucket and no two pairs contain four points from just two buckets then this represents a cubic graph on n vertices with no loops and no multiple edges. With probability bounded below by a positive constant, loops and multiple edges do not occur (see, for example, [13, Section 2.2]) . The buckets represent the vertices of the randomly generated cubic graph and each pair represents an edge whose end-points are given by the buckets of the points in the pair.

We may consider the generation process as follows. Initially, all vertices have degree 0. Throughout the execution of the generation process, vertices will increase in degree until the generation is complete and all vertices have degree 3. During this process, we refer to the graph being generated as the *evolving graph*.

3 Analysis Using Differential Equations

One method of analysing the performance of a randomised algorithm is to use a system of differential equations to express the expected changes in variables describing the state of the algorithm during its execution. Wormald [14] gives an exposition of this method and Duckworth [3] applies this method to various other graph-theoretic optimisation problems.

In order to analyse our algorithm using a system of differential equations, we incorporate the algorithm as part of a pairing process that generates a random cubic graph. In this way, we generate the random graph in the order that the edges are examined by the algorithm.

During the generation of a random cubic graph, we choose the pairs sequentially. The first point, p_i, of a pair may be chosen by any rule, but in order to ensure that the cubic graph is generated u.a.r., the second point, p_j, of that pair must be selected u.a.r. from all the remaining free points. The freedom of choice of p_i enables us to select it u.a.r. from the vertices of given degree in the evolving graph. Using $B(p_k)$ to denote the bucket that the point p_k belongs to, we say that the edge $(B(p_i), B(p_j))$ is *exposed* and this allows us to determine the degree of the vertex represented by the bucket $B(p_j)$.

The algorithm we use to find an induced matching of cubic graphs is a greedy algorithm based on selecting vertices of given degree. We say that our algorithm proceeds as a series of *operations*. An operation is the process of selecting an edge to add to the induced matching and the deletion of other edges. For each operation, a vertex v is chosen u.a.r. from those of given degree. An edge incident with v is selected to be added to the induced matching based on the degree(s) of the neighbour(s) of v. Other edges are then deleted in order to ensure that, after the next selection of an induced matching edge, the matching remains induced. Incorporating this as part of a pairing process that generates a random cubic graph, we select a vertex, v, u.a.r. from those of given degree in the evolving graph, expose its incident edges and investigate the degree(s) of its neighbour(s). An edge incident with v is selected to be added to the induced matching based on the degree(s) of the neighbour(s) of v. Further edges are then exposed in order to ensure the matching remains induced. More detail is given in the following section.

In what follows, we denote the set of vertices of degree i of the evolving graph by V_i and let Y_i $(=Y_i(t))$ denote $|V_i|$ (at some stage of the algorithm (time t)). We can express the state of the evolving graph at any point during the execution of the algorithm by considering Y_0, Y_1 and Y_2. In order to analyse our randomised algorithm for finding an induced matching of cubic graphs, we calculate the expected change in this state over one unit of time (a unit of time is defined more clearly in Section 5) in relation to the expected change in the size of the induced matching. Let M $(=M(t))$ denote the size of the induced matching at any stage of the algorithm (time t) and let $\mathbf{E}(\Delta \mathrm{X})$ denote the expected change in a random variable X conditional upon the history of the process. We then regard $\mathbf{E}(\Delta Y_i)/\mathbf{E}(\Delta M)$ as the derivative $\mathrm{d}Y_i/\mathrm{d}M$, which gives a system of differential equations. The solutions to these equations describe functions which represent the behaviour of the variables Y_i. There is a general result which guarantees that the solutions of the differential equations almost surely approximate the variables Y_i. The expected size of the induced matching may be deduced from these results.

4 The Algorithm

The degree of a vertex v in the evolving graph is denoted by $deg(v)$. We denote the set of all free points by P and use $q(b)$ to denote the set of free points in a given bucket b. The incorporated algorithm and pairing process, RANDMIM, is given in Figure 1; a description is given below.

```
select u u.a.r. from V_0;
select p_1 u.a.r. from q(u);
select p_2 u.a.r. from P;
v ← B(p_2);
M ← (u, v);
isolate(u, v);
while (Y_1 + Y_2 > 0)
do
   if (Y_2 > 0)
            select u u.a.r. from V_2;
            {p_1} ← q(u);
            select p_2 u.a.r. from P;
            v ← B(p_2);
   else
            select u u.a.r. from V_1;
            {p_1, p_2} ← q(u);
            select p_3 u.a.r. from P;
            a ← B(p_3);
            select p_4 u.a.r. from P;
            b ← B(p_4);
            if       (deg(a) > deg(b))   v ← a;
            else if  (deg(b) > deg(a))   v ← b;
            else     select v u.a.r. from {a, b};
   M ← M ∪ (u, v);
   isolate(u, v);
```

Figure 1: Algorithm RANDMIM

The function **isolate**(u, v) involves the process of exposing all the remaining edges incident with the vertices corresponding to the the buckets u and v and then exposing all remaining edges incident with the neighbours of u and v. This ensures that the matching remains induced.

The first operation of the algorithm involves randomly selecting the first edge of the induced matching and exposing the appropriate edges. We split the remainder of the algorithm into two distinct phases. We informally define Phase 1 as the period of time where any vertices in V_2 that are created are used up almost immediately and Y_2 remains small. Once the rate of generating vertices in V_2 becomes larger than the rate that they are used up, the algorithm moves into Phase 2 and all operations involve selecting a vertex from V_2. Note that the algorithm terminates when there are no remaining vertices of degree 1 or 2, which means that a connected component has been completely generated and a maximal induced matching has been found in that component. It is well known that cubic graphs are a.a.s. connected, so the result is a.a.s. a maximal induced matching in the whole graph.

There are two basic types of operation performed by the algorithm. A Type 1 operation refers to an operation where Y_2=0 and a vertex is chosen from V_1. Similarly, a Type 2 operation refers to an operation where $Y_2 > 0$ and a vertex is chosen from V_2. For Type 1, if (after exposing the edges incident with the chosen vertex, u, from V_1) exactly one of the neighbours, v, of u has degree 2, we add the edge (u, v) to the induced matching. Otherwise, we randomly choose an edge to add to the induced matching from those edges incident with u. For Type 2, we add to the induced matching, the edge incident with the chosen vertex, u, from V_2.

Figures 2 and 3 show the configurations that may be encountered by performing operations of Type 1 and Type 2 respectively (a.a.s.). The larger circles represent buckets each containing 3 smaller circles representing the points of that bucket. Smaller circles coloured black (respectively white) represent points that were, without a doubt, free (respectively used up) at the start of the operation. Smaller circles coloured grey represent points that were not known to be free or used up at the start of the operation.

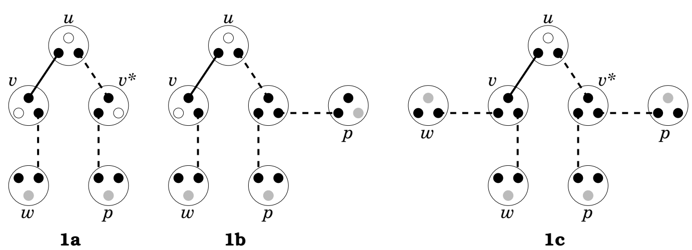

Figure 2: Type 1 operations

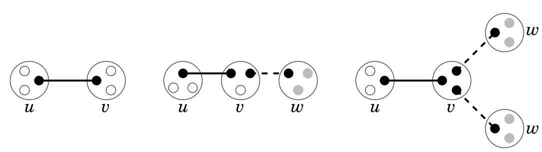

Figure 3: Type 2 operations

In all cases, the selected vertex is labelled u and the other end-point of the induced matching edge chosen is labelled v. A vertex labelled v^* denotes that a random choice has been made between 2 vertices and this one was not selected. After selecting a vertex u of given degree, the edges incident with this vertex are exposed. Once we determine the degrees of the neighbours of this vertex, we then make the choice as to which edge to add to the induced matching. Only then are other edges exposed. Therefore, at the start of the operation, we do not know the degrees of all the vertices at distance at most two from the end-points of the selected induced matching edge. A vertex whose degree is unknown is labelled either w or p. A vertex labelled p will have one of its incident edges exposed and will subsequently have its degree increased by one. We refer to these vertices as *incs* (as its degree is *inc*remented). A vertex labelled w will have all of its incident edges exposed and we refer to these vertices as *rems* (as they are *rem*oved from the set V_i). Should any *rem* be incident with other vertices of unknown degree, then these vertices will be *incs*.

Once an induced matching edge, e, has been selected, all edges incident with the end-points of e are exposed and subsequently all edges incident with the neighbours of the end-points of e are exposed.

5 The Lower Bound

Theorem 1 *For a random cubic graph on n vertices, the size of a maximum induced matching is asymptotically almost surely greater than* $0.270413n$.

Proof We define a *clutch* to be a series of operations in Phase i involving an operation of Type i and all subsequent operations up to but not including the next operation of Type i. Increment time by 1 unit for each clutch. We calculate $\mathbf{E}(\Delta Y_i)$ and $\mathbf{E}(\Delta M)$ for a clutch in each Phase.

5.1 Preliminary Equations For Phase 1

The initial opertion of Phase 1 is of Type 1 (at least a.a.s.). We consider operations of Type 2 first and then combine the equations given by these operations with those given by the operations of Type 1.

Operations of Type 2 involve the selection of a vertex u from V_2 (which has been created from processing a vertex from V_1). Let s $(=s(t))$ denote the number of free points available in all buckets at a given stage (time t). Note that $s=\sum_{i=0}^{2}(3-i)Y_i$. For our analysis it is convenient to assume that $s > \epsilon n$ for some arbitrarily small but fixed $\epsilon > 0$.

The expected change in Y_i due to changing the degree of an *inc* from i to $i+1$ by exposing one of its incident edges (at time t) is $\rho_i + o(1)$ where

$$\rho_i = \rho_i(t) = \frac{(i-3)Y_i + (4-i)Y_{i-1}}{s}, \quad 0 \le i \le 2$$

and this equation is valid under the assumption that $Y_{-1}=0$. To justify this, note that when the point in the *inc* was chosen, the number of points in the buckets corresponding to vertices currently of degree i is $(3-i)Y_i$, and s is the total number of points. In this case Y_i decreases; it increases if the selected point is from a vertex of degree $i-1$. These two quantities are added because expectation is additive. The term $o(1)$ comes about because the values of all these variables may change by a constant during the course of the operation being examined. Since $s > \epsilon n$ the error is in fact $O(1/n)$.

The expected change in Y_i due to exposing all edges incident with a *rem* and its incident *incs* (at time t) is $\mu_i + o(1)$ where

$$\mu_i = \mu_i(t) = \frac{(i-3)Y_i}{s} + \frac{(6Y_0 + 2Y_1)\rho_i}{s}, \quad 0 \le i \le 2.$$

The first term represents the removal of the *rem* from V_i (due to increasing its degree to 3). The expected number of *incs* incident with a *rem* is $(6Y_0 + 2Y_1)/s + o(1)$ and each of these will have its degree increased by 1 (giving the second term).

The expected change in Y_i for an operation of Type 2 in Phase 1 (at time t) is $\alpha_i + o(1)$ where

$$\alpha_i = \alpha_i(t) = \frac{(i-3)Y_i}{s} + \frac{(6Y_0 + 2Y_1)\mu_i}{s} - \delta_{i2}, \quad 0 \le i \le 2$$

in which

$$\delta_{ij} = \begin{cases} 1 & \text{if } \ i = j \\ 0 & \text{otherwise.} \end{cases}$$

We now consider operations of Type 1. The expected change in Y_i for operation $1h$ given in Figure 2 (at time t) is $\beta_{h,i} + o(1)$ where

$$\begin{aligned}
\beta_{a,i} &= \beta_{a,i}(t) = -3\delta_{i1} + \mu_i + \rho_i, \quad 0 \le i \le 2, \\
\beta_{b,i} &= \beta_{b,i}(t) = -\delta_{i0} - 2\delta_{i1} + \mu_i + 2\rho_i \quad 0 \le i \le 2 \quad \text{and} \\
\beta_{c,i} &= \beta_{c,i}(t) = -2\delta_{i0} - \delta_{i1} + 2\mu_i + 2\rho_i, \quad 0 \le i \le 2.
\end{aligned}$$

For an operation of Type 1 in Phase 1, the neighbours of u (the vertex selected at random from V_1) were in $\{V_0 \cup V_1\}$ before the start of the operation, since Y_2=0 when the algorithm performs this type of operation. The probability that these neighbours were in V_0 or V_1 are asymptotically $3Y_0/s$ and $2Y_1/s$ respectively. Therefore, the probabilities that, given we are performing an operation of Type 1 in Phase 1, the operation is of type 1a, 1b or 1c are given by

$$\begin{aligned}
\mathbf{P}(1a) &= \tfrac{4Y_1^2}{s^2} + o(1), \\
\mathbf{P}(1b) &= \tfrac{12Y_0Y_1}{s^2} + o(1) \ \text{ and} \\
\mathbf{P}(1c) &= \tfrac{9Y_0^2}{s^2} + o(1)
\end{aligned}$$

respectively.

We define a *birth* to be the generation of a vertex in V_2 by processing a vertex of V_1 or V_2 in Phase 1. The expected number of births from processing a vertex from V_1 (at time t) is $\nu_1 + o(1)$ where

$$\nu_1 = \nu_1(t) = \mathbf{P}(1a)\left(\mu_2 + \frac{2Y_1}{s}\right) + \mathbf{P}(1b)\left(\mu_2 + \frac{4Y_1}{s}\right) + \mathbf{P}(1c)\left(2\mu_2 + \frac{4Y_1}{s}\right).$$

Here, for each case, we consider the probability that vertices of degree 1 (in the evolving graph) become vertices of degree 2 by exposing an edge incident with the vertex.

Similarly, the expected number of births from processing a vertex from V_2 (at time t) is $\nu_2 + o(1)$ where

$$\nu_2 = \nu_2(t) = \frac{(6Y_0 + 2Y_1)\mu_2}{s}.$$

Consider the Type 1 operation at the start of the clutch to be the first generation of a *birth-death* process in which the individuals are the vertices in V_2, each giving birth to a number of children (essentially independent of the others) with expected number ν_2. Then, the expected number in the j^{th} generation is $\nu_1 {\nu_2}^{j-1}$ and the expected total number of births in the clutch is $\nu_1/(1-\nu_2)$.

For Phase 1, the equation giving the expected change in Y_i for a clutch is therefore given by

$$\mathbf{E}(\Delta Y_i) = \mathbf{P}(1a)\beta_{a,i} + \mathbf{P}(1b)\beta_{b,i} + \mathbf{P}(1c)\beta_{c,i} + \frac{\nu_1}{1-\nu_2}\alpha_i + o(1). \tag{1}$$

This assumes $Y_1 + Y_2$ is not zero, an eventuality which will be discussed later. The equation giving the expected increase in M for a clutch in Phase 1 is given by

$$\mathbf{E}(\Delta M) = 1 + \frac{\nu_1}{1-\nu_2} + o(1) \tag{2}$$

since the contribution to the increase in the size of the induced matching by the Type 1 operation in a clutch is 1 and for each birth we have a Type 2 operation (a.a.s.).

5.2 Preliminary Equations For Phase 2

In Phase 2, all operations are considered to be of Type 2 and therefore a clutch consists of one operation. The expected change in Y_i is given by

$$\mathbf{E}(\Delta Y_i) = \alpha_i + o(1)$$

where α_i remains the same as that given for Phase 1 and the expected increase in M is 1 per clutch.

5.3 The Differential Equations

The equation representing $\mathbf{E}(\Delta Y_i)$ for processing a clutch in Phase 1 forms the basis of a differential equation. Write $Y_i(t)=nz_i(t/n)$, $\mu_i(t)=n\tau_i(t/n)$, $\beta_{j,i}(t)=n\psi_{j,i}(t/n)$, $s(t)=n\xi(t/n)$, $\alpha_i(t)=n\chi_i(t/n)$ and $\nu_j(t)=n\omega_j(t/n)$. The differential equation suggested is

$$z_i' = \frac{4z_1^2}{\xi^2}\psi_{a,i} + \frac{12z_0z_1}{\xi^2}\psi_{b,i} + \frac{9z_0^2}{\xi^2}\psi_{c,i} + \frac{\omega_1}{1-\omega_2}\chi_i, \quad 0 \le i \le 2 \tag{3}$$

where differentiation is with respect to x and xn represents the number, t, of clutches. From the definitions of μ, β, s, α and ν we have

$$\begin{aligned}
\tau_i &= \tfrac{(i-3)}{\xi}z_i + \tfrac{(6z_0+2z_1)((i-3)z_i+(4-i)z_{i-1})}{\xi^2}, \quad 0 \le i \le 2, \\
\psi_{a,i} &= -3\delta_{i1} + \tau_i + \tfrac{(i-3)z_i+(4-i)z_{i-1}}{\xi}, \quad 0 \le i \le 2, \\
\psi_{b,i} &= -\delta_{i0} - 2\delta_{i1} + \tau_i + 2\tfrac{(i-3)z_i+(4-i)z_{i-1}}{\xi}, \quad 0 \le i \le 2, \\
\psi_{c,i} &= -2\delta_{i0} - \delta_{i1} + 2\tau_i + 2\tfrac{(i-3)z_i+(4-i)z_{i-1}}{\xi}, \quad 0 \le i \le 2, \\
\xi &= \textstyle\sum_{i=0}^{2}(3-i)z_i, \\
\chi_i &= \tfrac{(i-3)}{\xi}z_i + \tfrac{6z_0+2z_1}{\xi}\tau_i - \delta_{i2}, \quad 0 \le i \le 2, \\
\omega_1 &= \tfrac{4z_1^2}{\xi^2}\left(\tau_2 + \tfrac{2z_1}{\xi}\right) + \tfrac{12z_0z_1}{\xi^2}\left(\tau_2 + \tfrac{4z_1}{\xi}\right) + \tfrac{4z_1^2}{\xi^2}\left(2\tau_2 + \tfrac{4z_1}{\xi}\right) \quad \text{and} \\
\omega_2 &= \tfrac{6z_0+2z_1}{\xi}\tau_2.
\end{aligned}$$

Using the equation representing the expected increase in the size M of $\mathcal{M}$ after processing a clutch in Phase 1 and writing $M(t)=nz(t/n)$ suggests the differential equation for z as

$$z' = 1 + \frac{\omega_1}{1-\omega_2}. \tag{4}$$

For Phase 2 the equation representing $\mathbf{E}(\Delta Y_i)$ for processing a clutch suggests the differential equation

$$z_i' = \chi_i, \quad 0 \le i \le 2. \tag{5}$$

The solution to these systems of differential equations represents the cardinalities of the sets V_i and $\mathcal{M}$ (scaled by $\frac{1}{n}$) for given t. For Phase 1, the equations are (3) and (4) with initial conditions

$$z_0(0) = 1, \quad z_i(0) = 0 \ \ (i > 0).$$

The initial conditions for Phase 2 are given by the final conditions for Phase 1, and the equations are given by (5).

In [14] is a general result which we use to show that during each phase, the functions representing the solutions to the differential equations almost surely approximate the random variables Y_i/n and M/n with error $o(1)$. For arbitrarily small $\epsilon > 0$, define D_1 to be the set of all (t, z_0, z_1, z_2, z) for which $t > -\epsilon$, $\xi > \epsilon$, $\omega_2 < 1 - \epsilon$, $z > -\epsilon$ and $z_i < 1 + \epsilon$ where $0 \leq i \leq 2$. D_1 defines a domain for the variables t, z_i and z so that [14, Theorem 6.1] may be applied to the process within Phase 1. Equations (1) and (2) verify the trend hypothesis of [14, Theorem 5.1], which is also used in [14, Theorem 6.1]. (Note in particular that since $\xi > \epsilon$ inside D_1, the assumption that $s > \epsilon n$ used in deriving these equations is justified.) For [14, Theorem 6.1] we may also eliminate a set of undesirable states, which we characterise by $Y_1 + Y_2 \leq 0$. The conclusion is that the random variables Y_i/n and M/n a.a.s. remain within $o(1)$ of the corresponding deterministic solutions to the differential equations (3) and (4) until a point arbitrarily close to where it leaves the domain D_1, or an undesirable state is achieved. Since the latter can only occur when the algorithm has completely processed a component of the graph, and a random cubic graph is a.a.s. connected, we may turn to examining the former.

We compute the ratio $\frac{dz_i}{dz} = \frac{z_i'(x)}{z'(x)}$, and we have

$$\frac{dz_i}{dz} = \frac{\frac{4z_1^2}{\xi^2}\psi_{a,i} + \frac{12z_0z_1}{\xi^2}\psi_{b,i} + \frac{9z_0^2}{\xi^2}\psi_{c,i} + \frac{\omega_1}{1-\omega_2}\chi_i}{1 + \frac{\omega_1}{1-\omega_2}}, \quad 0 \leq i \leq 2,$$

where differentiation is with respect to z and all functions can be taken as functions of z. By solving (numerically) this system of differential equations, we find that the solution hits a boundary of the domain at $\omega_2 = 1 - \epsilon$ (for $\epsilon = 0$ this would be at time $t \geq 0.134887n$). Formally, Phase 1 is defined as the period of time from time $t = 0$ to the time t_0 representing the solution of $\omega_2 = 1$.

An argument similar to that given for independent sets in [14] or that given for independent dominating sets in [5] ensures that a.a.s. the process passes through phases as defined informally, and that Phase 2 follows Phase 1.

Once in Phase 2, vertices in V_2 are replenished with high probability which keeps the process in Phase 2. For Phase 2 and for arbitrary small ϵ, define D_2 to be the set of all (t, z_0, z_1, z_2, z) for which $t > t_0 + \epsilon$, $\xi > \epsilon$, $z > -\epsilon$ and $z_i < 1 + \epsilon$ where $0 \leq i \leq 2$. Theorem 6.1 from [14] applies as in Phase 1 except that here, a clutch consists of just one operation. Note that the increase in the size of the induced matching per clutch processed in Phase 2 is 1, so computing the ratio $\frac{dz_i}{dz} = \frac{z_i'(x)}{z'(x)}$ gives

$$\frac{dz_i}{dz} = \chi_i, \quad 0 \leq i \leq 2.$$

By solving this we see that the solution hits a boundary of D_2 at $\xi = \epsilon$ (for $\epsilon = 0$ this would be approximately $0.270413n$).

The differential equations were solved using a Runge-Kutta method, giving $\omega_2 = 1$ at $z \geq 0.134887$ and in Phase 2, $z_2 = 0$ at $z > 0.270413$. This corresponds to the size of the induced matching (scaled by $\frac{1}{n}$) when all vertices are used up, thus proving the theorem. $\square$

6 The Upper Bound

We now establish an upper bound on the size of a maximum induced matching of a random cubic graph.

Theorem 2 *For a random cubic graph on n vertices, the size of a maximum induced matching is asymptotically almost surely less than* $0.282069n$.

Proof Consider a random n-vertex cubic graph G generated using the pairing model given in Section 2. Let $M(G,k,s)$ denote the number of *maximal* induced matchings of G of size k (where s is the number of vertices in the set S of vertices adjacent to the end-points of the matching edges). We calculate $\mathbf{E}(M(G,k,s))$ and show that when $k = 0.282069n$, $\mathbf{E}(M(G,k,s)) = o(1)$, for every choice of S, thus proving the theorem. Let $N(i) = \frac{(2i)!}{i!2^i}$.

Given a maximal induced matching of size k we have a set K of $2k$ vertices that are the end-points of the matching edges, a set S of s vertices that are adjacent to the end-points of the matching edges and a set R of the remaining $n-2k-s$ vertices. By maximality, R forms an independent set.

The number of ways to choose the set K, the set S and the points in the buckets corresponding to the end-points of the $2k$ matching edges is

$$\binom{n}{2k}\binom{n-2k}{s}N(k)3^{2k}.$$

Denote the number of ways to choose the $4k$ edges each incident with a vertex in K and a vertex in S by $a(k,s)$. For each vertex in S with j points matched, these can be chosen in $\binom{3}{j}$ ways. Hence the number of ways of doing one such choice for each vertex in S has ordinary generating function $f(x)^s$ where

$$f(x) = \sum_{j=1}^{3}\binom{3}{j}x^j = (1+x)^3 - 1.$$

This implies

$$a(k,s) = (4k)![x^{4k}]((1+x)^3-1)^s$$

where the square brackets mean taking the coefficient.

The number of ways to pair the $3(n-2k-s)$ points of R with points from the remaining $3s-4k$ free points in S is

$$\frac{(3s-4k)!}{(6s+2k-3n)!}$$

and the number of ways to complete the pairing is

$$\frac{(6s+2k-3n)!}{(3s+k-\frac{3n}{2})!2^{3s+k-\frac{3n}{2}}}.$$

The total number of pairings is given by $N(3n/2)$.

Combining these expressions, $\mathbf{E}$(M(G,k,s)) is given by

$$\frac{\binom{n}{2k}\binom{n-2k}{s}N(k)3^{2k}(4k)![x^{4k}]((1+x)^3-1)^s(3s-4k)!N(3s+k-\frac{3n}{2})}{(6s+2k-3n)!N(\frac{3n}{2})}$$

which simplifies to

$$\mathbf{E}(M(G,k,s)) = \frac{n!3^{2k}(4k)![x^{4k}]((1+x)^3-1)^s(3s-4k)!(\frac{3n}{2})!2^{3n-2k-s}}{(n-2k-s)!s!k!(3s+k-\frac{3n}{2})!(3n)!}.$$

As $[x^u]g(x) \le \frac{g(x)}{x^u}$ for all positive real values of x this implies

$$\mathbf{E}(M(G,k,s)) \le \frac{n!3^{2k}(4k)!((1+x)^3-1)^s(3s-4k)!(\frac{3n}{2})!2^{3n-2k-3s}}{(n-2k-s)!s!k!x^{4k}(3s+k-\frac{3n}{2})!(3n)!}.$$

Approximate using Stirling's formula and re-write using $f(y) = y^y$, $\kappa = k/n$ and $\gamma = s/n$. We have

$$\mathbf{E}(M(G,k,s))^{\frac{1}{n}} \sim \frac{3^{2\kappa} f(4\kappa)((1+\hat{x})^3 - 1)^{\gamma} f(3\gamma - 4\kappa) f(\frac{3}{2}) 2^{3-2\kappa-3\gamma}}{f(1-2\kappa-\gamma) f(\gamma) f(\kappa) \hat{x}^{4\kappa} f(3\gamma + \kappa - \frac{3}{2}) f(3)}, \tag{6}$$

where $\hat{x} = \hat{x}(k,s)$ denotes the value that minimises this quantity and it may be verified that this is given by the solution to the equation

$$\frac{4k}{s} = \frac{3x(x+1)^2}{(x+1)^3 - 1}.$$

Using $\frac{3-2\kappa}{6} \leq \gamma \leq 1 - 2\kappa$, we find that for $\kappa \geq 0.282069$ the expression on the right of (6) tends to 0. A small amount of work is required to prove that this is indeed the maximum. This may be achieved by computing the derivative with respect to x. $\square$

References

[1] Bollobás, B.: *Random Graphs.* Academic Press, London, 1985.

[2] Cameron, K.: Induced Matchings. *Discrete Applied Mathematics*, 24:97–102, 1989.

[3] Duckworth, W.: *Greedy Algorithms and Cubic Graphs.* PhD thesis, Department of Mathematics and Statistics, University of Melbourne, Australia. *Submitted.*

[4] Duckworth, W., Manlove, D. and Zito, M.: On the Approximability of the Maximum Induced Matching Problem. Technical Report TR-2000-56 of the Computing Science Department, Glasgow University, UK, April 2000.

[5] Duckworth, W. and Wormald, N.C.: Minimum Independent Dominating Sets of Random Cubic Graphs. *Submitted.*

[6] Erdös, E.: Problems and Results in Combinatorial Analysis and Graph Theory. *Discrete Mathematics*, 72:81–92, 1988.

[7] Faudree, R.J., Gyárfas, A., Schelp, R.H. and Tuza, Z.: Induced Matchings in Bipartite Graphs. *Discrete Mathematics*, 78:83–87, 1989.

[8] Golumbic, M.C. and Laskar, R.C.: Irredundancy in Circular Arc Graphs. *Discrete Applied Mathematics*, 44:79–89, 1993.

[9] Golumbic, M.C. and Lewenstein, M.: New Results on Induced Matchings. *Discrete Applied Mathematics*, 101:157–165, 2000.

[10] Liu, J. and Zhou, H.: Maximum Induced Matchings in Graphs. *Discrete Mathematics*, 170:271–281, 1997.

[11] Steger, A. and Yu, M.: On Induced Matchings. *Discrete Mathematics*, 120:291–295, 1993.

[12] Stockmeyer, L.J. and Vazirani, V.V.: NP-Completeness of Some Generalisations of the Maximum Matching Problem. *Information Processing Letters*, 15(1):14–19, 1982.

[13] Wormald, N.C.: Models of random regular graphs. *Surveys in combinatorics, 1999 (Canterbury)*, 239–298, Cambridge University Press, Cambridge, 1999.

[14] Wormald, N.C.: Differential Equations for Random Processes and Random Graphs. In *Lectures on Approximation and Randomized Algorithms*, 73–155, PWN, Warsaw, 1999. Michał Karoński and Hans-Jürgen Prömel (editors).

[15] Zito, M.: Induced Matchings in Regular Graphs and Trees. In *Proceedings of the 25th International Workshop on Graph Theoretic Concepts in Computer Science*, volume 1665 of Lecture Notes in Computer Science , 89–100. Springer-Verlag, 1999.

[16] Zito, M.: *Randomised Techniques in Combinatorial Algorithmics*. PhD thesis, Department of Computer Science, University of Warwick, UK, November 1999.