Maximum Induced Matchings of Random Cubic Graphs

William Duckworth[1], Nicholas C. Wormald[1*], and Michele Zito[2**]

[1] Department of Mathematics and Statistics, University of Melbourne, Australia
{billy,nick}@ms.unimelb.edu.au
[2] Department of Computer Science, University of Liverpool, UK
michele@csc.liv.ac.uk

Abstract. In this paper we present a heuristic for finding a large induced matching $\mathcal{M}$ 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 returned by the algorithm. The corresponding upper bound is derived by means of a direct expectation argument. We prove that $\mathcal{M}$ asymptotically almost surely satisfies $0.2704n \le |\mathcal{M}| \le 0.2821n$.

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 $\mathcal{M}$ are connected by an edge of $E \setminus \mathcal{M}$. We are interested in finding induced matchings of large cardinality.

Stockmeyer and Vazirani [11] 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) [5, 6, 9, 10]).

The problem of deciding whether for a given integer k a given graph G has an induced matching of size at least k is NP-Complete [11], even for bipartite graphs of maximum degree 4. It has been shown [3, 13] that the problem is APX-complete even when restricted to ks-regular graphs for $k \in \{3, 4\}$ and any integer $s \ge 1$. The problem of finding a maximum induced matching is polynomial-time solvable for chordal graphs [2] and circular arc graphs [7].

* Supported by the Australian Research Council
** Supported by EPSRC grant GR/L/77089

Recently Golumbic and Lewenstein [8] 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 $\mathcal{M}$ 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 returned by the algorithm. The corresponding upper bound is derived by means of a direct expectation argument. We prove that $\mathcal{M}$ asymptotically almost surely satisfies $0.2704n \leq |\mathcal{M}| \leq 0.2821n$. Little is known on the complexity of this problem under the additional assumption that the input graphs occur with a given probability distribution. Zito [14] presented some simple results on dense random graphs. The algorithm we present was analysed deterministically in [3] where it was shown that, given an n-vertex cubic graph, the algorithm returns an induced matching of size at least $3n/20 + o(1)$, and there exist infinitely many cubic graphs realising 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). 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 Uniform Generation of Random Cubic Graphs

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 perfect matching of the $3n$ points.

If no edge of the matching contains two points from the same bucket and no two edges 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 parallel edges do not occur [1]. The edges of the random cubic graph generated are represented by the edges of the matching and its vertices are represented by the buckets.

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. We refer to the graph being generated by this process as the *evolving graph*.

3 Analysing Algorithms Using Differential Equations

We incorporate the algorithm as part of a matching process generating a random cubic graph. During the generation of a random cubic graph, we choose the matching edges sequentially. The first end-point of a matching edge may be chosen by any rule, but in order to ensure that the cubic graph is generated u.a.r., the second end-point of that edge must be selected u.a.r. from all the remaining free points. The freedom of choice of the first end-point of a matching edge enables us to select it u.a.r. from the vertices of given degree in the evolving graph.

The algorithm we use to generate an induced matching of cubic graphs is a greedy algorithm based on choosing a vertex of a given degree and performing some edge and vertex deletions. In order to analyse our algorithm using a system of differential equations, we generate the random graph in the order that the edges are examined by the algorithm.

At some stage of the algorithm, a vertex is chosen of given degree based upon the number of free points remaining in each bucket. The remaining edges incident with this vertex are then *exposed* by randomly selecting a mate for each spare point in the bucket. This allows us to determine the degrees of the neighbours of the chosen vertex and we refer to this as *probing* those vertices. This is done without exposing the other edges incident with these probed vertices. Once the degrees of the neighbours of the chosen vertex are known, an edge is selected to be added to the induced matching. Further edges are then exposed in order to ensure the matching is 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 time t). The number of edges in the induced matching at any stage of the algorithm (time t) is denoted by $M(= M(t))$ and we let $s(= s(t))$ denote the number of free points available in buckets at any stage of the algorithm (time t). Note that $s = \sum_{i=0}^{2}(3-i)Y_i$. Let $\mathbf{E}(\Delta \mathrm{X})$ denote the expected change in a random variable X conditional upon the history of the process.

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 an algorithm during its execution (see [12] for an exposition of this method).

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 time step in relation to the expected change in the size M of the induced matching. 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 solution to these equations describes functions which represent the behaviour of the variables Y_i. There is a general result which guarantees that the solution of the differential equations almost surely approximates the variables Y_i. The expected size of the induced matching may be deduced from these results.

4 The Algorithm

In order to find an induced matching of a cubic graph, we use the following algorithm. We assume the generated graph G to be connected (since random cubic graphs are connected a.a.s.). Then, at any stage of the algorithm after the first step and before its completion, $Y_1 + Y_2 > 0$. The degree of a vertex v in the evolving graph is denoted by $deg(v)$. We denote the initial set of $3n$ points by P. $B(p)$ denotes the bucket that a given point p belongs to and we use $q(b)$ to denote the set of free points in bucket b. The set of induced matching edges returned is denoted by $\mathcal{M}$. Here is the algorithm; a description is given below.

```
select(p1, P);
expose(p1, p2);
isolate(B(p1), B(p2));
M ← (B(p1), B(p2));
while (Y1 + Y2 > 0)
 { if (Y2 > 0)
     { select(u, V2);  {p1} ← q(u);
       expose(p1, p2);  v ← B(p2);}
   else
     { select(u, V1);  {p1, p2} ← q(u);
       expose(p1, p3);  a ← B(p3);
       expose(p2, p4);  b ← B(p4);
       if       (deg(a) > deg(b))  v ← a;
       else if  (deg(b) > deg(a))  v ← b;
       else                        select(v, {a, b});}
   isolate(u, v);
   M ← M ∪ (u, v);
 }
```

The function **select**(s, S) involves the process of selecting the element s u.a.r. from the set S. The function **expose**(p_i, p_j) involves the process of deleting the selected point p_i from P, exposing the edge $(B(p_i), B(p_j))$ by randomly selecting the point p_j from P and deleting the point p_j from P. The function **isolate**(B_1, B_2) involves the process of randomly selecting a mate for each free point in the buckets B_1 and B_2 and then exposing all edges incident with these selected mates. This ensures that the matching is induced.

The first step 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 the majority of operations involve selecting a vertex from V_2 and including its incident edge in the induced matching.

There are two basic operations performed by the algorithm. Type 1 refers to a step when $Y_2 > 0$ and a vertex is chosen from V_2. Similarly, Type 2 refers to a step where $Y_2 = 0$ and a vertex is chosen from V_1.

Figs. 1 and 2 show the configurations that may be encountered performing operations of Type 1 and Type 2 respectively (a.a.s.). For Type 1, we add the edge incident with the chosen vertex. For Type 2, if (after exposing the edges incident with the chosen vertex u) exactly one of the neighbours of u has exactly two free points, we add the edge incident with this vertex and the chosen vertex. Otherwise we randomly choose an edge incident with the chosen vertex.

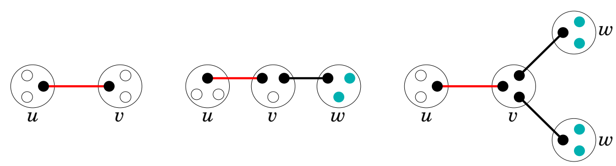

Fig. 1. Selecting a vertex from V_2 and adding its incident edge to $\mathcal{M}$.

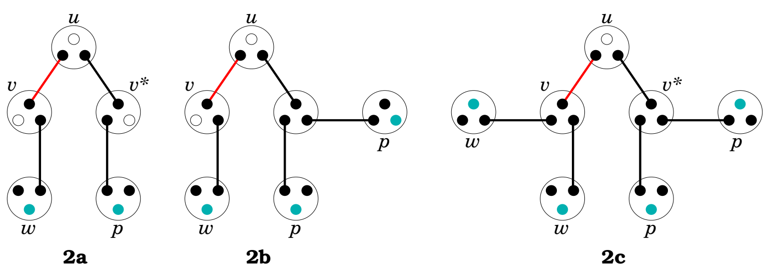

Fig. 2. Selecting a vertex from V_1 and choosing an edge to add to $\mathcal{M}$.

The larger circles represent buckets each containing 3 smaller circles representing the points of that bucket. Smaller circles coloured black represent points that are free. Used points are represented by white circles. Points which are not known to be free at this stage of the algorithm are shaded.

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 probe 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 this stage, 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*. A vertex labelled w will have all of its incident edges exposed and we refer to these vertices as *rems*. Should any *rem* be incident with other vertices of unknown degree then these vertices will be *incs*.

Once the choice of induced matching edge has been made, all edges incident with the endpoints of the induced matching edge are exposed and subsequently the edges of the neighbours of the end-points of the chosen edge are exposed. This ensures the matching is induced.

5 The Lower Bound

Theorem 1. *For a cubic graph on n vertices the size of a maximum induced matching is asymptotically almost surely greater than* $0.2704n$

Proof. We define a *clutch* to be a series of operations in Phase i involving the selection a vertex from V_i and all subsequent operations up to but not including the next selection of a vertex from V_i. Increment time by 1 step for each clutch of vertices processed. 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 step of Phase 1 is of Type 2 (at least a.a.s.). We consider operations of Type 1 first and then combine the equations given by these operations with those given by the operations of Type 2.

Operations of Type 1 involve the selection of a vertex u from V_2 (which has been created from processing a vertex from V_1). The expected change in Y_2 is negligible and can assumed to be 0 since this is phase 1 and Y_2 remains small as vertices of V_2 are used up almost as fast as they are created. The desired equation may be formulated by considering the contribution to $\mathbf{E}(\Delta Y_i)$ in three parts; deleting v (the neighbour of u), deleting the neighbour(s) of v and increasing the degree of the other neighbours of these neighbours by one.

Let ρ_i denote the contribution to $\mathbf{E}(\Delta Y_i)$ due to changing the degree of an *inc* from i to $i+1$ and we have

$$\rho_i = \frac{(i-3)Y_i + (4-i)Y_{i-1}}{s}.$$

This equation is valid under the assumption that $Y_{-1} = 0$.

We let $\mu_i = \mu_i(t)$ denote the contribution to $\mathbf{E}(\Delta Y_i)$ due to a *rem* and all its edges incident with *incs* and we have

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

Let $\alpha_i = \alpha_i(t)$ denote the contribution to $\mathbf{E}(\Delta Y_i)$ for an operation of Type 1 in Phase 1. We have

$$\alpha_i = \frac{(i-3)}{s} Y_i + \left(\frac{6Y_0 + 2Y_1}{s}\right) \mu_i.$$

We now consider operations of Type 2. Let $\beta_{h,i}(=\beta_{h,i}(t))$ denote the contribution to $\mathbf{E}(\Delta Y_i)$ for operation $2h$ given in Fig. 2 (at time t). We will also use $\delta_{ij}=1$ if $i=j$ and $\delta_{ij}=0$ otherwise.

We have

$$\beta_{a,i} = -3\delta_{i1} + \mu_i + \rho_i,$$

$$\beta_{b,i} = -\delta_{i0} - 2\delta_{i1} + \mu_i + 2\rho_i,$$

$$\beta_{c,i} = -2\delta_{i0} - \delta_{i1} + 2\mu_i + 2\rho_i.$$

These equations are formulated by considering the contribution to $\mathbf{E}(\Delta Y_i)$ by deletion of vertices of known degree and the contribution given by the expected degree of vertices of unknown degree.

For an operation of Type 2 in Phase 1, neighbours of u (the vertex selected at random from V_1) are in $\{V_0 \cup V_1\}$, since $Y_2 = 0$ when the algorithm performs this type of operation. The probability that these neighbours are in V_0 or V_1 are $3Y_0/s$ and $2Y_1/s$ respectively. Therefore the probabilities that, given we are performing an operation of Type 2 in Phase 1, the operation is of type 2a, 2b or 2c are given by $\mathbf{P}(2a) = \frac{4Y_1^2}{s^2}$, $\mathbf{P}(2b) = \frac{12Y_0Y_1}{s^2}$ and $\mathbf{P}(2c) = \frac{9Y_0^2}{s^2}$ respectively.

We define a birth to be the generation of a vertex in V_2 by processing a vertex of V_1. Let $\nu_1(=\nu_1(t))$ be the expected number of births from processing a vertex from V_1 (at time t). Then we have

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

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

Similarly, we let $\nu_2 = \nu_2(t)$ be the expected number of births from processing a vertex from V_2. Then we have

$$\nu_2 = \left(\frac{6Y_0 + 2Y_1}{s}\right)\mu_2,$$

giving the expected number of births in a clutch to be $\nu_1/(1-\nu_2)$.

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

$$\mathbf{E}(\Delta Y_i) = \mathbf{P}(2a)\beta_{a,i} + \mathbf{P}(2b)\beta_{b,i} + \mathbf{P}(2c)\beta_{c,i} + \frac{\nu_1}{1-\nu_2}\alpha_i.$$

The equation giving the expected increase in M for a clutch is given by

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

since the contribution to the increase in the size of the induced matching by the Type 2 operation in a clutch is 1.

5.2 Preliminary Equations For Phase 2

In Phase 2, all operations are considered to be of Type 1 and therefore a clutch consists of one step, but we must also consider the expected change in Y_2 since this is no longer a negligible amount. The expected change in Y_i is given by

$$\mathbf{E}(\Delta Y_i) = \alpha_i - \delta_{i2}$$

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

5.3 The Differential Equations

The equation representing $\mathbf{E}(\Delta Y_i)$ for processing a clutch of vertices 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

$$z_i' = \frac{4z_1^2}{\xi^2}\psi_{a,i} + \frac{12z_0z_1}{\xi^2}\psi_{b,i} + \frac{9z_0^2}{\xi^2}\psi_{c,i} + \frac{\omega_1}{1-\omega_2}\chi_i \qquad (i \in \{0,1\})$$

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

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

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

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

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

$$z_i' = \frac{\frac{4z_1^2}{\xi^2}\psi_{a,i} + \frac{12z_0z_1}{\xi^2}\psi_{b,i} + \frac{9z_0^2}{\xi^2}\psi_{c,i} + \frac{\omega_1}{1-\omega_2}\chi_i}{1 + \frac{\omega_1}{1-\omega_2}} \qquad (i \in \{0,1\})$$

where differentiation is with respect to z and all functions can be taken as functions of z.

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

$$z_i' = \chi_i - \delta_{i2} \qquad (0 \le i \le 2). \tag{1}$$

The increase in the size of the induced matching per clutch of vertices processed in this Phase is 1, so computing the ratio $\frac{dz_i}{dz} = \frac{z_i'(x)}{z'(x)}$ gives the same equation as that given in (1). Again, differentiation is with respect to z and all functions can be taken as functions of z.

The solution to this system of differential equations represents the cardinalities of the sets V_i for given M (scaled by $\frac{1}{n}$). For Phase 1, the initial conditions are

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

The initial conditions for Phase 2 are given by the final conditions for Phase 1.

Wormald [12] describes a general result which ensures that the solutions to the differential equations almost surely approximate the variables Y_i. It is simple to define a domain for the variables z_i so that Theorem 5.1 from [12] may be applied to the process within each phase. An argument similar to that given for independent sets in [12] or that given for independent dominating sets in [4] ensures that a.a.s. the process passes through phases as defined informally, and that Phase 2 follows Phase 1. Formally, Phase 1 ends at the time corresponding to $\omega_2 = 1$ as defined by the equations for Phase 1. Once in Phase 2, vertices in V_2 are replenished with high probability which keeps the process in Phase 2.

The differential equations were solved using a Runge-Kutta method, giving $\omega_2 = 1$ at $z = 0.1349$ and in Phase 2 $z_2 = 0$ at $z > 0.2704$. This corresponds to the size of the induced matching (scaled by $\frac{1}{n}$) when all vertices are used up, thus proving the theorem. □

6 The Upper Bound

Theorem 2. *For a cubic graph on n vertices the size of a maximum induced matching is asymptotically almost surely less than* $0.2821n$

Proof. Consider a random cubic graph G on n vertices. Let $M(G,k)$ denote the number of induced matchings of G of size k. We calculate $\mathbf{E}(M(G,k))$ and show that when $k > 0.2821n$, $\mathbf{E}(M(G,k)) = o(1)$ thus proving the theorem. Let $N(x) = \frac{(2x)!}{x!2^x}$.

$$\begin{aligned}\mathbf{E}(M(G,k)) &\leq \binom{n}{2k} N(k)\, 3^{2k}\, \frac{(3(n-2k))!}{(3n-10k)!}\, \frac{N(\frac{3n}{2}-5k)}{N(\frac{3n}{2})}\\ &= \frac{n!\, 3^{2k}\, (3(n-2k))!\, (\frac{3n}{2})!\, 2^{4k}}{k!\, (n-2k)!\, (\frac{3n}{2}-5k)!\, (3n)!}.\end{aligned}$$

Approximate using Stirling's formula and re-write using $f(x)=x^x$, $\kappa = k/n$ and we have

$$\mathbf{E}(M(G,k))^{\frac{1}{n}} \sim \frac{3^{2\kappa}\, f(3(1-2\kappa))\, f(\frac{3}{2})\, 2^{4\kappa}}{f(\kappa)\, f(1-2\kappa)\, f(\frac{3}{2}-5\kappa)\, f(3)}.$$

Solving this we find that for $\kappa \geq 0.2821$ the expression on the right tends to 0. □

Note that this bound may be improved by counting only maximal matchings. However the improvement is slight and we do not include the details here for reasons of brevity.

References

1. Bollobás, B.: Random Graphs. Academic Press, London, 1985.
2. Cameron, K.: Induced Matchings. Discrete Applied Math., **24**:97–102, 1989.
3. Duckworth, W., Manlove, D. and Zito, M.: On the Approximability of the Maximum Induced Matching Problem. J. of Combinatorial Optimisation (Submitted).
4. Duckworth, W. and Wormald, N.C.: Minimum Independent Dominating Sets of Random Cubic Graphs. Random Structures and Algorithms (Submitted).
5. Erdös, E.: Problems and Results in Combinatorial Analysis and Graph Theory. Discrete Math., **72**:81–92, 1988.
6. Faudree, R.J., Gyárfas, A., Schelp, R.H. and Tuza, Z.: Induced Matchings in Bipartite Graphs. Discrete Math., **78**:83–87, 1989.
7. Golumbic, M.C. and Laskar, R.C.: Irredundancy in Circular Arc Graphs. Discrete Applied Math., **44**:79–89, 1993.
8. Golumbic, M.C. and Lewenstein, M.: New Results on Induced Matchings. Discrete Applied Math., **101**:157–165, 2000.
9. Liu, J. and Zhou, H.: Maximum Induced Matchings in Graphs. Discrete Math., **170**:271–281, 1997.
10. Steger, A. and Yu, M.: On Induced Matchings. Discrete Math., **120**:291–295, 1993.
11. Stockmeyer, L.J. and Vazirani, V.V.: NP-Completeness of Some Generalizations of the Maximum Matching Problem. Inf. Proc. Lett., **15**(1):14–19, 1982.
12. 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).
13. 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.
14. Zito, M.: Randomised Techniques in Combinatorial Algorithmics. PhD thesis, Department of Computer Science, University of Warwick, UK, November 1999.