

# Controllability and matchings in random bipartite graphs

Paul Balister and Stefanie Gerke

## Abstract

Motivated by an application in controllability we consider maximum matchings in random bipartite graphs  $G = (A, B)$ . First we analyse Karp–Sipser’s algorithm to determine the asymptotic size of maximum matchings in random bipartite graphs with a fixed degree distribution. We then allow an adversary to delete one edge adjacent to every vertex in  $A$  in the more restricted model where each vertex in  $A$  chooses  $d$  neighbours uniformly at random from  $B$ .

## 1 Introduction

We are interested in finding large matchings in random bipartite graphs. The motivation comes in part from recent work by Liu, Slotine, and Barabási [16], in which they used a characterisation by Lin [15] of structural controllability to show how large matchings in random bipartite graphs play a crucial role in obtaining bounds on the number of nodes needed to control directed networks. We will give a short description of this connection in Section 2.

Matchings in bipartite graphs are a classical problem in graph theory. The famous theorem of Hall [10] states that a bipartite graph with vertex sets  $V_1$  and  $V_2$  contains a matching of size  $|V_1|$  if and only if for every set  $S \subseteq V_1$  we have  $|S| \leq |\Gamma(S)|$  where  $\Gamma(S)$  is the neighbourhood of  $S$ . One can use this characterisation to show that in a random bipartite graph  $G(n, n, p)$  with vertex sets  $V_1$  and  $V_2$  of the same size  $n$  where each of the possible  $n^2$  edges is present with probability  $p$  independent of the presences or absence of all other edges, with high probability there is a matching of size  $n$  if there is no isolated vertex. The random bipartite graph  $G(n, n, p)$  has no isolated vertices with high probability if  $np - \log n$  tends to infinity as  $n$  tends to infinity, see for example [11].

Finding matchings in general graphs is a well-studied problem and polynomial time algorithms are known to find maximum matchings, see for example [7, 17]. We will analyse a far simpler algorithm developed by Karp and Sipser [14] which we will introduce in Section 3.3. This algorithm is known to work well asymptotically almost surely on sparse random graphs [1]. Bohman and Frieze [4] analysed this algorithm for the class of graphs which have no vertices of degree smaller than  $\delta \geq 2$

or larger than  $\Delta$ , and  $\delta n_\delta, \dots, \Delta n_\Delta$  forms a log-concave sequence. Here  $n_d$  is the number of vertices of degree  $d$ . We generalise this result in Section 3 to more general degree distributions and develop a simpler proof. Bohman and Frieze used the differential equation method to track the number of vertices of each degree whereas we use generating functions and will see that we only need to track two variables. Non-algorithmic proofs for slightly more precise results can be found in [5] and also in [18]. In particular the authors determine the size of a maximum matching in a random bipartite graph with a fixed degree distribution under some mild assumption on these distributions.

Motivated by the application in controllability we discuss briefly in Section 4 the presence of an adversary in a more restricted model. More precisely, we consider a random bipartite graph  $G = (A \cup B, E)$  with  $|A| = n$  and  $|B| = (1 + \varepsilon)n$ . Each vertex in  $A$  is adjacent to  $d$  neighbours chosen uniformly at random from  $B$ . We allow repetition, so this is a multigraph. An adversary is then able to remove a single edge adjacent to each vertex of  $A$ , with the aim of minimising the size of the largest matching. The complete proofs can be found in [2]. Melsted and Frieze [9] analysed the Karp–Sipser algorithm for the random bipartite graph where each vertex in a partition class of size  $n$  chooses  $d$  neighbours uniformly at random from a partition class of size  $\alpha n$  for some  $\alpha > 0$  but without an adversary.

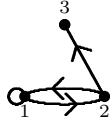
## 2 Controllability

Roughly speaking, controllability is concerned with which elements of a network one needs to be able to manipulate — a control set — to be able to force the entire network into any desired state. More precisely, we have a vector  $\mathbf{x} \in \mathbb{R}^n$  that evolves according to a system of equations  $d\mathbf{x}(t)/dt = \mathbf{f}(\mathbf{x}) + \mathbf{y}(t)$  where  $\mathbf{y} \in \mathbb{R}^n$  is a ‘controlling’ term with  $y_i(t) = 0$  whenever  $i$  is not in the control set. We are interested in the smallest control set  $S$  such that by varying  $y_i$  for  $i \in S$  suitably we can force  $\mathbf{x}$  into any desired state in finite time. We consider the simpler case when the changes are linear. We have a vector  $\mathbf{x}(t) = (x_1(t), x_2(t), \dots, x_n(t))$  which represents the current state of a system with  $n$  nodes at time  $t$ ; the  $n \times n$  matrix  $A$  showing the topology of the system signalling interactions between nodes and the  $n \times m$  matrix  $B$  that identifies the set of nodes controlling the system. Thus we are interested in  $d\mathbf{x}(t)/dt = A\mathbf{x}(t) + B\mathbf{u}(t)$ . The canonical controllability criterion according to Kalman [12] is that the  $n \times nm$  matrix  $C = (B, AB, A^2B, \dots, A^{n-1}B)$  has full rank. In applications it is often impossible to measure the entries of  $A$  exactly or they may be time dependent (for example internet traffic). Structural Control-

libility circumvents this problem by allowing us to choose the non-zero entries in  $A$  and  $B$  such that  $C = (B, AB, A^2B, \dots, A^{n-1}B)$  has maximal rank. It can be shown [15, 19] that a system that is structurally controllable is controllable for almost all choices of the entries, except for some pathological cases of zero measure that occur when the system parameters satisfy certain accidental constraints. Structural controllability has applicability in many areas beyond classical control systems, including in large-scale networks found in national critical infrastructures such as energy and telecommunication networks, and where such networks interact, as e.g., in Smart Grid control systems [6].

From a graph-theoretic point of view structural controllability is attractive because one can interpret the problem as a directed graph problem: first one observes that if one can choose the entries freely one can always assume that the entries  $b_{ij}$  of  $B$  are 0 if  $i \neq j$  and 1 if  $i = j$ . Hence one only needs to consider  $A$  which can be represented by a directed graph  $G$  with a directed arc between a vertex  $i$  and  $j$  if  $a_{ij} \neq 0$ , i.e., if variable  $x_i$  affects  $dx_j/dt$ . For example:

$$\begin{aligned} \frac{dx_1}{dt} &= ax_1 + bx_2 + y_1 \\ \frac{dx_2}{dt} &= cx_1 \\ \frac{dx_3}{dt} &= dx_2 \end{aligned}$$



$$S = \{1\}$$

Equations

Graph  $G$ 

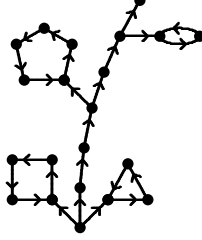
Controlling set

There are fairly simple obstructions for a set  $S$  to control a network:

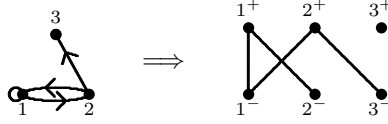
**Inaccessibility:** There is a vertex  $v$  such that there is no directed path from any vertex  $u \in S$  to  $v$ .

**Dilation:** There is a subset of vertices  $X$ ,  $X \cap S = \emptyset$ , such that the size of the in-neighbourhood is smaller than  $|X|$ .

Lin [15] showed that these necessary conditions are indeed sufficient, that is, if there are no inaccessible vertices and no dilation then the set  $S$  controls the network. He also showed that the number of nodes needed to control a system is equal to the minimum number of vertex-disjoint *cacti* that span  $G$ . To define a cactus we need to define stems and buds first. A *stem* is a simple directed path. The first vertex is called the root and the last vertex is called the top of the stem. A *bud* is a simple directed cycle with an additional edge that ends but does not begin in a vertex of the cycle. A cactus consists of a stem and a collection of disjoint buds such that the initial vertex of the additional edge of each bud belongs to the stem. We may also assume that this initial vertex is not the top vertex. The following is an example of a cactus.



Consider the undirected bipartite graph  $B$  obtained from a directed graph  $G$  by splitting each vertex  $v$  into  $v^+$  and  $v^-$  and adding an edge between  $u^+$  and  $v^-$  in  $B$  if there is a directed edge  $uv$  in  $G$ .



Note that a matching  $M$  in  $B$  corresponds to a union of  $|V(G)| - |M|$  (possibly trivial) directed paths and some number of directed cycles in  $G$ . But in a sparse random graph there are whp only  $o(n)$  directed cycles as the expected number of directed cycles of length  $k$  is constant for each fixed  $k$ . Hence if one wants to find the minimum number of disjoint cacti that span a sparse random graph up to an  $o(n)$  term then one is interested in a maximum matching in  $B$ . Indeed, a vertex cover by  $t$  cacti gives rise to a vertex cover of  $t$  directed paths and  $o(n)$  directed cycles, while a cover by  $t$  directed paths and  $s = o(n)$  directed cycles gives rise to a cover by  $t + s$  directed paths, which can be considered as  $t + s$  cacti.

### 3 Maximum matchings in random bipartite graphs with a fixed degree distribution

#### 3.1 The random graph models

Fix two sequences  $(z_i)_{i=0}^{\infty}$  and  $(\hat{z}_i)_{i=0}^{\infty}$  such that  $z_i, \hat{z}_i \geq 0$  and

$$0 < \mu := \sum_{i=0}^{\infty} i z_i = \sum_{i=0}^{\infty} i \hat{z}_i < \infty. \quad (3.1)$$

We consider a sequence of random bipartite (multi-)graphs  $B_n = (V_n \cup \hat{V}_n, E_n)$  with bipartite classes  $V_n$  and  $\hat{V}_n$  such that

$$|V_n| = (1 + o(1))n \sum_{i=0}^{\infty} z_i, \quad |\hat{V}_n| = (1 + o(1))n \sum_{i=0}^{\infty} \hat{z}_i,$$

by fixing the degrees of the vertices in  $V_n$  and  $\hat{V}_n$  in such a way that as  $n$  tends to infinity the number of vertices of degree  $i$  in  $V_n$  is  $(1 + o(1))nz_i$  and the number of vertices of degree  $i$  in  $\hat{V}_n$  is  $(1 + o(1))n\hat{z}_i$ . Naturally, we require that the sum of the degrees in  $V_n$  and  $\hat{V}_n$  are the same. For technical reasons we assume that the total number of edges is  $(1 + o(1))n\mu$ . This assumption is mainly to avoid examples such as a sequence of double stars which have average degree 2 but all but two vertices have degree 1 and the size of the maximum matching is 2. Note that we do not require that  $|V_n| = |\hat{V}_n|$  nor do we require that the distributions  $(z_i)$  and  $(\hat{z}_i)$  are the same.

Having fixed the degrees  $d(v)$  of the vertices  $v \in V_n \cup \hat{V}_n$ , we choose the edges randomly using the configuration model, that is, each vertex  $v$  is replaced by the appropriate number of *configuration points*  $v_1, \dots, v_{d(v)}$  and a perfect matching is chosen uniformly at random over all perfect matchings between the set of configuration points of  $V$  and the configuration points of  $\hat{V}$ . We then identify the configuration points corresponding to each vertex to obtain the bipartite multi-graph  $B_n$  with vertex classes  $V$  and  $\hat{V}$ .

We will also consider the corresponding non-bipartite version  $G_n = (V_n, E_n)$  where we have a single degree distribution  $(z_i)_{i=0}^\infty$ ,  $|V_n| = (1 + o(1))n \sum z_i$ , and the total number of configuration points is chosen to be even and asymptotically  $(1 + o(1))n\mu$ , where  $\mu = \sum iz_i < \infty$ .

We are interested in the size of a maximum matching up to an  $o(n)$  error. Because of the error term we may assume that the degrees are in fact bounded by an absolute constant  $\Delta$ . To see this, note that adding or removing a single edge from a graph can affect the size of the maximum matching by at most 1. Since our distributions have finite mean we have that for all  $\varepsilon > 0$ , we can choose  $\Delta$  sufficiently large such that

$$\sum_{i=0}^{\Delta} iz_i \geq \mu - \varepsilon.$$

Let  $n_i^{(n)}$  be the number of vertices of degree  $i$  in  $V_n$ . Then, since  $|E_n| = (1 + o(1))n\mu$  (in the bipartite case), we have that for sufficiently large  $n$

$$\sum_{i=0}^{\Delta} in_i^{(n)} \geq n(\mu - 2\varepsilon) \geq |E_n| - 3\varepsilon n.$$

Therefore there are at most  $3\varepsilon n$  edges in  $B_n$  that are incident to vertices in  $V_n$  of degree more than  $\Delta$ . By choosing  $\Delta$  sufficiently large we can assume the same for  $\hat{V}_n$ . Hence removing at most  $6\varepsilon n$  edges results in a

graph of maximum degree at most  $\Delta$  with a degree distribution that is asymptotically  $6\epsilon$  close to the original distribution. Moreover, it has a maximum matching that is within  $6\epsilon n$  of the original graph. A similar argument holds in the non-bipartite case of  $G_n = (V_n, E_n)$ . Thus we may assume for the remainder of the paper that the degrees are bounded by an absolute constant  $\Delta$  which is independent of  $n$ . Similarly, by modifying the degrees of  $o(n)$  vertices we may in fact assume that there are *exactly*  $z_i n$  vertices in  $V_n$  and  $\hat{z}_i n$  vertices in  $\hat{V}_n$  of degree  $i$ . (Assuming of course that the  $z_i, \hat{z}_i$  are rational,  $n$  is a multiple of their denominators, and  $\sum i z_i n$  is even in the non-bipartite case.) To simplify notation we shall usually drop the subscript  $n$  and write the graphs as  $B = (V \cup \hat{V}, E)$  or  $G = (V, E)$ .

### 3.2 Results

We consider the following generating functions. Let

$$f(x) = \sum_{i=0}^{\Delta} z_i x^i \quad \text{and} \quad \hat{f}(x) = \sum_{i=0}^{\Delta} \hat{z}_i x^i.$$

Note that  $|V| = f(1)n$ ,  $|\hat{V}| = \hat{f}(1)n$  and  $\mu = f'(1) = \hat{f}'(1)$ , so that  $|E| = f'(1)n = \hat{f}'(1)n$ .

Let  $w_1, w_2, \hat{w}_1, \hat{w}_2$  be the smallest non-negative solutions to the following simultaneous equations

$$w_1 = \frac{f'(\hat{w}_2)}{f'(1)}, \quad w_2 = 1 - \frac{f'(1-\hat{w}_1)}{f'(1)}, \quad \hat{w}_1 = \frac{\hat{f}'(w_2)}{\hat{f}'(1)}, \quad \hat{w}_2 = 1 - \frac{\hat{f}'(1-w_1)}{\hat{f}'(1)}. \quad (3.2)$$

Define

$$\begin{aligned} \xi &= f(1) + \hat{f}(1) - f(\hat{w}_2) - \hat{f}(1-w_1) - \hat{f}'(1-w_1)w_1, \\ \hat{\xi} &= f(1) + \hat{f}(1) - \hat{f}(w_2) - f(1-\hat{w}_1) - f'(1-\hat{w}_1)\hat{w}_1. \end{aligned} \quad (3.3)$$

**Theorem 3.1** *Suppose that  $\xi \leq \hat{\xi}$  and that*

$$\frac{\beta f''(\alpha)}{f'(\alpha + \beta) - f'(\alpha)} \cdot \frac{\hat{\beta} \hat{f}''(\hat{\alpha} + \hat{\beta})}{\hat{f}'(\hat{\alpha} + \hat{\beta}) - \hat{f}'(\hat{\alpha})} \leq 1 \quad (3.4)$$

*along the trajectory  $(\alpha, \beta, \hat{\alpha}, \hat{\beta})$  starting at  $(\hat{w}_2, 1-\hat{w}_1-\hat{w}_2, w_2, 1-w_1-w_2)$  and evolving according to the differential equations given in (3.9) up until the point at which  $\phi = \sqrt{\delta}$ . Then with probability tending to 1 as  $n \rightarrow \infty$ , the size of a maximum matching in  $B = (V \cup \hat{V}, E)$  is*

$$(\xi - O(\sqrt{\delta}))n.$$

The conditions in Theorem 3.1 are somewhat awkward to check, however a much simpler statement is possible in the symmetric case when  $z_i = \hat{z}_i$ .

**Theorem 3.2** *Suppose that  $z_i = \hat{z}_i$  for all  $i$ , so that  $f = \hat{f}$ ,  $w_i = \hat{w}_i$ , and  $\xi = \hat{\xi}$ , and suppose that  $f''(x)^{-1/2}$  is convex on  $[0, 1]$ . Then with probability tending to 1 as  $n \rightarrow \infty$ , the size of a maximum matching in  $B = (V \cup \hat{V}, E)$  is*

$$(\xi - o(1))n.$$

This last result also generalises to the non-bipartite case.

**Theorem 3.3** *Let  $f(x) = \sum_i z_i x^i$  be the generating function of  $(z_i)_{i=0}^\infty$ . Suppose that  $f''(x)^{-1/2}$  is convex on  $[0, 1]$ . Define  $w_1$  and  $w_2$  to be the smallest non-negative solutions to the following simultaneous equations*

$$w_1 = \frac{f'(w_2)}{f'(1)}, \quad w_2 = 1 - \frac{f'(1-w_1)}{f'(1)}.$$

*Let  $\xi = 2f(1) - f(w_2) - f(1-w_1) - f'(1-w_1)w_1$ . Then with probability tending to 1 as  $n \rightarrow \infty$ , the size of a maximum matching in  $G = (V, E)$  is*

$$(\xi - o(1))n.$$

Let us remark that the condition that  $f''(x)^{-1/2}$  is convex (or equivalently  $2f''(x)f^{(4)}(x) \leq 3f^{(3)}(x)^2$ ) holds for a wide variety of distributions, in particular it is implied by the log-concavity of  $dz_d$  which was assumed in [4]. We will discuss this condition later.

### 3.3 Karp–Sipser’s algorithm

Given a graph  $G$ , Karp–Sipser’s algorithm is a randomized algorithm that starts with an empty graph  $G'$  on the same vertex set, and chooses at each step an edge  $e$  of  $G$ , adds  $e$  to the graph  $G'$  and deletes  $e$  and all edges incident to  $e$  from  $G$ . If there are vertices of degree 1 in  $G$  then Karp–Sipser’s algorithm chooses an edge incident to a vertex of degree 1 independently at random from all those edges. If there are no vertices of degree 1 then the Karp–Sipser’s algorithm chooses an edge uniformly at random from all remaining edges of the graph.

For ease of exposition, we will delete the end vertices of  $e$  from the graph  $G$  when  $e$  is added to the matching. We shall say that a vertex *becomes isolated* if its degree is reduced to zero *without* it being included in the matching.

One key property of the Karp–Sipser algorithm is that it is optimal as long as there are degree 1 vertices, as given any degree 1 vertex  $v$

meeting the edge  $e$ , there is always some maximum matching containing the edge  $e$ . We split the Karp–Sipser algorithm into two phases. The first phase is all steps of the algorithm that occur before the first time when there are no degree 1 vertices left. The second phase is the remaining steps of the algorithm. As noted, the first phase is always optimal, so we can bound the difference between the size of the matching  $M_{\text{KS}}$  produced by the Karp–Sipser algorithm and the size of a maximum matching  $M_{\text{max}}$ . Indeed, suppose after the first phase the graph  $G$  has  $N$  remaining non-isolated vertices. If  $N_0$  of these  $N$  vertices are not matched in the second phase of Karp–Sipser, then

$$|M_{\text{KS}}| \leq |M_{\text{max}}| \leq |M_{\text{KS}}| + \lfloor N_0/2 \rfloor.$$

Indeed, the best we can hope for in the second phase is an additional  $\lfloor N/2 \rfloor$  edges in our matching, so if we have  $(N - N_0)/2$  additional edges then we are at most  $\lfloor N_0/2 \rfloor$  short of a maximum matching. In the bipartite case we can say even more. Suppose there after the first phase the graph  $B$  has  $N$  remaining non-isolated vertices in the *smaller* bipartite class. If  $N_0$  of these  $N$  vertices are not matched in the second phase of Karp–Sipser, then

$$|M_{\text{KS}}| \leq |M_{\text{max}}| \leq |M_{\text{KS}}| + N_0,$$

as the maximum possible matching of the remaining edges would contain at most  $N$  edges.

Another key property of the Karp–Sipser algorithm is that at every step, if we condition on the degrees of the remaining vertices, the edges are distributed according to the configuration model. Indeed, we need only reveal the configuration as necessary as the algorithm proceeds. So, for example, when we choose a random edge we choose two configuration points uniformly at random and when we determine the neighbour of a particular configuration point, it is chosen uniformly at random from the collection of remaining configuration points. This gives the correct probability distribution of choices at each stage as in a uniformly chosen perfect matching, choosing a random edge is equivalent to picking two configuration points uniformly at random, and each configuration point is matched to another configuration point with a uniform distribution. Also, conditioning on the existence of an edge, the perfect matching of the remaining configuration points has a uniform distribution.

In Section 3.4 it will be convenient to modify the algorithm slightly so that when there are several vertices of degree 1 we do not necessarily choose one uniformly at random, but instead choose them according to some other scheme. The argument above still applies however, as long as the choice of degree 1 vertex does not depend on the edge it is incident



to (i.e., on the choice of its matching configuration point). In Section 3.5 we will modify the algorithm further, allowing mistakes where an edge is chosen uniformly at random from the graph even when a small number of degree 1 vertices exist. Once again, the above argument is unaffected as long as the algorithm depends only on the degrees of the remaining vertices and not on the specific edges between them.

### 3.4 The first phase of Karp–Sipser

To analyse the first phase of the Karp–Sipser algorithm we will use branching-process techniques. It will turn out that the finite neighbourhood of most vertices is a tree and therefore branching processes are a good approximation. We will analyse the first phase of the algorithm in rounds. A round consist of marking all the degree 1 vertices and then processing each of these in some order. (The actual order is arbitrary, as long as it does not depend on the edges incident to these vertices.) Note that some of the degree 1 vertices may become isolated before they are processed in which case they are simply ignored in subsequent rounds. Note that strictly speaking this is a slight modification of Karp–Sipser, as we insist that all vertices of degree 1 in round  $t$ , say, are processed before any vertex of degree 1 that is generated by the removal of edges in round  $t$ .

Consider a vertex  $v$  and all vertices at distance at most  $t$  from  $v$ . The vertex  $v$  is called *t-good* or *good* if its  $t$ -neighbourhood is a tree. Otherwise it is *bad*. The next lemma shows that most vertices are good when  $t = o(\log n)$ .

**Lemma 3.4** *Assume that  $t = o(\log n)$ . Then with high probability the number of bad vertices is  $o(n)$ .*

**Proof** We just give the proof for the bipartite case as non-bipartite case is similar. Fix a vertex  $v_1$  and consider its  $t$ -neighbourhood. If  $v_1$  is bad then this neighbourhood must contain a cycle. Pick a shortest such cycle and a shortest path connecting this cycle to  $v_1$ . This way we obtain a path  $v_1, v_2, \dots, v_\ell$  such that  $\ell \leq 2t + 1$  and  $v_\ell$  is adjacent to some  $v_j$ ,  $j < \ell - 1$ , or doubly adjacent to  $v_{\ell-1}$ . We count the expected number of such configurations. The probability of one such configuration is bounded from above by

$$\left( \prod_{i=1}^{\ell-1} \frac{d(v_i)d(v_{i+1})}{|E| - i + 1} \right) \frac{d(v_\ell)d(v_j)}{|E| - \ell + 1} \leq \frac{\Delta^{2\ell}}{(|E| - \ell)^\ell} \leq \frac{\Delta^{2\ell}}{|E|^{\ell-1}(|E| - \ell^2)}.$$

Indeed, conditioning on the presence of particular edges from  $v_k$  to  $v_{k+1}$

for  $k < i$ , the probability of an edge from  $v_i$  to  $v_{i+1}$  is at most

$$\frac{d(v_i)d(v_{i+1})}{|E| - i + 1}$$

as for each of the  $d(v_i)$  (actually  $d(v_i) - 1$  for  $i > 1$ ) configuration points corresponding to  $v_i$ , there are at most  $d(v_{i+1})$  out of the remaining  $|E| - i + 1$  configuration points in the other bipartite class that this point can be joined to, and each are equally likely.

The number of such configurations is at most  $\ell|E|^{\ell-1}$  as there are at most  $|E|$  choices for each of the the vertices  $v_2, \dots, v_\ell$  and  $\ell$  choices for  $j$ . (Clearly isolated vertices are not valid choices for the  $v_i$  and so there at most  $|E|$  choices for each vertex.) Hence the probability of  $v_1$  being bad is at most  $\ell\Delta^{2\ell}/(|E| - \ell^2) \leq c^\ell/n$  for some  $c > 0$  as  $|E| \sim \mu n$  and  $\ell = o(\log n)$ . The result now follows from Markov's inequality as

$$\mathbb{P}[\text{number of bad vertices} \geq \alpha] \leq c^\ell/\alpha.$$

□

Note that the fate of a vertex  $v$  during the first  $k$  rounds of the first phase of Karp–Sipser's algorithm is completely determined by the  $2k + 1$  neighbourhood of  $v$ . For the remainder of this section we assume that the vertex  $v$  is  $t$ -good.

Fix a good vertex  $v$  and consider the tree of depth  $t + 1$  rooted at  $v$ . Inductively, we classify all vertices at distance at most  $t + 1$  other than  $v$  as either  $v$ -lonely,  $v$ -popular or  $v$ -normal with respect to  $t$ . At depth  $t + 1$  all vertices are  $v$ -normal. Suppose we have classified all vertices at level  $\ell + 1$ . Then a vertex at level  $\ell$  is  $v$ -lonely if all its children on level  $\ell + 1$  are  $v$ -popular. This also applies to the case when it has no children on level  $\ell + 1$ . A vertex is  $v$ -popular if at least one of its children is  $v$ -lonely. Otherwise a vertex is  $v$ -normal. The root  $v$  itself will be *lonely* with respect to  $t$  if all of its children are  $v$ -popular (including the case when it is isolated), it will be *popular* with respect to  $t$  if at least *two* of its children are  $v$ -lonely and *normal* with respect to  $t$  otherwise (see Figure 1).

**Lemma 3.5** *Assume  $v$  is a  $t$ -good vertex and  $t \geq 1$ .*

- (a) *If  $v$  is lonely with respect to  $t$  then it will either be matched or become isolated within the first  $\lfloor t/2 \rfloor$  rounds. If it is matched to  $u$  then  $u$  is popular with respect to  $t - 1$ .*
- (b) *If  $v$  is popular with respect to  $t$  then it will be matched within the first  $\lfloor t/2 \rfloor$  rounds. If it is matched to  $u$  then  $u$  is lonely with respect to  $t + 1$ .*

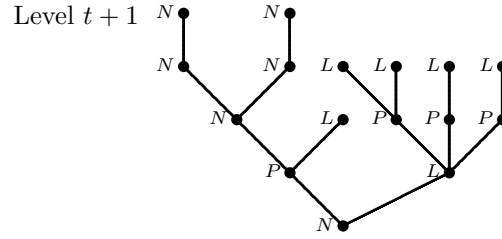


Figure 1: Classification of lonely, popular, and normal vertices.

(c) If  $v$  is normal with respect to  $t$  then it will not become isolated during the first  $\lfloor t/2 \rfloor$  rounds. If it is matched to another vertex  $u$  during the first  $\lfloor t/2 \rfloor$  rounds, then  $u$  is normal with respect to  $t - 1$ .

**Proof** Note first that whatever happens during the first  $\lfloor t/2 \rfloor$  rounds to vertex  $v$  only depends on the tree rooted at  $v$  of depth  $t+1$ . In particular whether a vertex becomes isolated or of degree 1 during the first  $\lfloor t/2 \rfloor$  rounds only depends on this tree.

We now observe that as long as no chosen matching edge contains the root  $v$ , all matching edges connect a  $v$ -lonely vertex (with respect to  $t$ ) with an  $v$ -popular parent (with respect to  $t$ ). Indeed, after  $i$  rounds, if  $v$  is not matched then any  $v$ -popular vertex at distance at least  $t - 2i$  from  $v$  will be matched with a child that is  $v$ -lonely, and any  $v$ -lonely vertex at distance at least  $t + 1 - 2i$  from  $v$  will be either matched or become isolated. Note that no vertex at level  $t$  can be  $v$ -popular, so any  $v$ -lonely vertex at distance  $t$  or  $t - 1$  from  $v$  has no children. Hence, assuming  $v$  is not matched or isolated after  $\lfloor t/2 \rfloor$  rounds, then all its  $v$ -popular children will have been matched, leaving it isolated if  $v$  is itself lonely. If  $v$  is lonely and matched, then it must have done so because it became degree 1. The vertex  $u$  that it is matched to is clearly popular with respect to  $t - 1$  as it is  $v$ -popular. Thus  $u$  has a  $v$ -lonely child, which is  $u$ -lonely with respect to  $t - 1$ . On the other hand, all the children of  $v$  other than  $u$  were matched after  $\lfloor t/2 \rfloor - 1$  rounds and so must be  $v$ -popular with respect to  $t - 2$ . Thus  $v$  is  $u$ -lonely with respect to  $t - 1$ . Hence  $u$  has two  $u$ -lonely children and so is popular with respect to  $t - 1$ .

Similarly, if  $v$  is popular, then at least two of its children are  $v$ -lonely. But all of their children are then matched within the first  $\lfloor t/2 \rfloor - 1$  rounds, and so these children are of degree 1 before round  $\lfloor t/2 \rfloor$ . Thus  $v$  is matched in this round if not before. Since  $v$  has two children that are  $v$ -lonely, and  $v$ -lonely vertices can only be matched to their parent prior to  $v$  being matched or isolated, it is clear  $v$  does not become isolated or degree 1,

but instead is matched to one of its children. This child  $u$  is clearly lonely with respect to  $t + 1$  as all its neighbours (including  $v$ ) are  $u$ -popular ( $v$  itself having a lonely child other than  $u$ ).

Now assume  $v$  is normal. Then not all of its children are  $v$ -popular and therefore not all of its children can be removed during this process. Therefore  $v$  never becomes isolated. Now assume  $v$  is matched to  $u$  and assume for a contradiction that  $u$  is not normal with respect to  $t - 1$ . Before  $v$  is matched, the only children it can lose are  $v$ -popular by the observation above. Moreover, these  $v$ -popular neighbours have degree at least 2 or are matched to an  $v$ -lonely child. Assume first that  $u$  is popular with respect to  $t - 1$ . Since  $v$  is normal there must be at least one neighbour  $w$  of  $v$  that is not  $v$ -popular with respect to  $t$  and therefore won't be removed before  $v$  is matched. This means  $v$  has degree at least 2 when it is matched. But  $u$  must have a  $u$ -lonely neighbour other than  $v$  that cannot be matched except to  $u$ . Hence  $u$  also has degree 2 or is matched to a  $u$ -lonely vertex which is not  $v$ . Therefore  $uv$  cannot be a matching edge.

Now assume that  $u$  is lonely with respect to  $t - 1$ . Then  $u$  is  $v$ -lonely with respect to  $t$ . Also  $v$  is  $u$ -popular with respect to  $t - 1$  which means that there is at least one child other than  $u$  which is  $u$ -lonely with respect to  $t - 1$  and in particular it is  $v$ -lonely with respect to  $t$  as being  $v$ -lonely is monotone in  $t$ . Thus  $v$  has two  $v$ -lonely children with respect to  $t$ , a contradiction to the assumption that  $v$  is normal with respect to  $t$ .  $\square$

We will show that the solutions  $w_1$ ,  $w_2$ ,  $\hat{w}_1$ , and  $\hat{w}_2$  of (3.2) mentioned in the introduction represent the probability of a vertex in  $V$  or  $\hat{V}$  being  $v$ -lonely or  $v$ -popular if they are close to a good vertex  $v$ . We use iterated versions of  $w_i$  and  $\hat{w}_i$  to make this precise. In particular, we define  $w_i^{(t)}$  and  $\hat{w}_i^{(t)}$  using the following recursion.

Set

$$w_1^{(0)} = w_2^{(0)} = \hat{w}_1^{(0)} = \hat{w}_2^{(0)} = 0,$$

and for  $t \geq 1$  inductively define

$$w_1^{(t)} = \frac{f'(\hat{w}_2^{(t-1)})}{f'(1)}, \quad w_2^{(t)} = 1 - \frac{f'(1 - \hat{w}_1^{(t-1)})}{f'(1)},$$

and

$$\hat{w}_1^{(t)} = \frac{\hat{f}'(w_2^{(t-1)})}{\hat{f}'(1)}, \quad \hat{w}_2^{(t)} = 1 - \frac{\hat{f}'(1 - w_1^{(t-1)})}{\hat{f}'(1)}. \quad (3.5)$$

**Theorem 3.6** *Let  $t = o(\log n)$  and fix a good vertex  $v$ . For all  $i = 1, \dots, t$  the following holds.*

- (a) *The probability that a vertex  $u$  in  $V$  at distance  $t + 1 - i$  is  $v$ -lonely with respect to  $t$  conditioned on its existence and the tree of depth  $t$*

from  $v$  except for the branch rooted at  $u$  is  $w_1^{(i)} + O(c^{t+i}/n)$  for some constant  $c$ .

- (b) The probability that a vertex  $u$  in  $V$  at distance  $t+1-i$  is  $v$ -popular with respect to  $t$  conditioned on its existence and the tree of depth  $t$  from  $v$  except for the branch rooted at  $u$  is  $w_2^{(i)} + O(c^{t+i}/n)$  for some constant  $c$ .

The corresponding statements hold for  $u \in \hat{V}$  with  $w_1^{(i)}$ ,  $w_2^{(i)}$  replaced with  $\hat{w}_1^{(i)}$ ,  $\hat{w}_2^{(i)}$  respectively. Moreover, these results remain true even conditioned on the location of  $O(c^t)$  edges outside of the tree of depth  $t$  from  $v$ .

**Proof** The statements are clearly true for  $i = 0$  as all vertices at level  $t+1$  are  $v$ -normal with respect to  $t$ .

Assume that we explored the entire tree with root  $v$  of depth  $t+1$  except for one branch starting at level  $t+1-i$ . We know that there is an edge from level  $t-i$  to a vertex  $u$  but we do not know anything further about this branch. Note that in the configuration model we have fixed at most  $\Delta^{t+1}$  edges. The probability that  $u$  is  $v$ -lonely (conditioning on the remaining tree) is

$$\sum_{k=0}^{\Delta-1} \mathbb{P}(u \text{ has children } u_1 \dots u_k) \prod_{j=1}^k \mathbb{P}(u_j \text{ is } v\text{-pop.} \mid u_1 \dots u_{j-1} \text{ are } v\text{-pop.}). \quad (3.6)$$

The probability that at the beginning, we choose an edge adjacent to vertex of degree  $k+1$  is  $(k+1)z_{k+1}/\mu$  and hence the probability that  $u$  has  $k$  children is

$$\frac{(k+1)z_{k+1}n - O(\Delta^t)}{|E| + O(\Delta^t)} = \frac{(k+1)z_{k+1}}{\mu} + O(\Delta^t/n).$$

Note that the  $O(\Delta^t)$  term takes care of the edges that are already fixed. The probability  $p_k$  that a child  $u_k$  of  $u$  is  $v$ -popular with respect to  $t$  conditioned on the remaining tree including the other children of  $u$  is by

induction hypothesis  $\hat{w}_2^{(i-1)} + O(c^{t+i-1}/n)$ . Thus (3.6) gives

$$\begin{aligned}
& \sum_{k=0}^{\Delta-1} \left( \frac{(k+1)z_{k+1}}{\mu} + O(\Delta^t/n) \right) p_k^k \\
&= \sum_{k=0}^{\Delta-1} \frac{(k+1)z_{k+1}}{\mu} p_k^k + O(\Delta^{t+1}/n) \\
&= \frac{f'(p_k)}{f'(1)} + O(\Delta^{t+1}/n) \\
&= \frac{f'(\hat{w}_2^{(i-1)})}{f'(1)} + \frac{f''(1)}{f'(1)} O(c^{t+i-1}/n) + O(\Delta^{t+1}/n).
\end{aligned}$$

Here we used that  $p_k = \hat{w}_2^{(i-1)} + O(c^{t+i-1}/n)$  is between 0 and 1 and that the maximum second derivative of  $f$  in  $[0, 1]$  is  $f''(1)$ . Note that the first term is by definition  $w_1^{(i)}$  and the error term is  $O(c^{t+1}/n)$  provided  $c > \max\{f''(1)/f'(1), \Delta\}$ .

The other statements are proved analogously.  $\square$

**Theorem 3.7** *Let  $t = t(n) \rightarrow \infty$  with  $t = o(\log n)$ . Then the following hold whp.*

- (a) *The number of good vertices  $v \in V$  that are lonely w.r.t.  $t$  is  $(f(\hat{w}_2) + o(1))n$ .*
- (b) *The number of good vertices  $v \in V$  that are popular w.r.t.  $t$  is  $(f(1) - f(1 - \hat{w}_1) - f'(1 - \hat{w}_1)\hat{w}_1 + o(1))n$ .*
- (c) *The number of good vertices  $v \in V$  that remain unmatched and non-isolated after  $t$  rounds is  $(f(1 - \hat{w}_1) - f(\hat{w}_2) - f'(\hat{w}_2)\hat{w}_2 + o(1))n$ .*
- (d) *The number of vertices  $v \in V$  that are of degree  $k \geq 1$  after  $t$  rounds is  $(z'_k + o(1))n$ , where  $z'_k$  are defined by the generating function  $\sum z'_k x^k = f(\alpha + \beta x) - f(\alpha) - \beta f'(\alpha)x$ , and  $\alpha = \hat{w}_2$ ,  $\beta = 1 - \hat{w}_1 - \hat{w}_2$ .*
- (e) *The number of vertices that have degree 1 after  $t$  rounds is  $o(n)$ .*

The same statements apply to  $\hat{V}$  with  $f$  replaced by  $\hat{f}$  and  $\hat{w}_i$  replaced by  $w_i$ .

**Proof** Note that  $w_i^{(t)}$  and  $\hat{w}_i^{(t)}$  are bounded increasing sequences and as  $t \rightarrow \infty$ ,  $w_i^{(t)} \rightarrow w_i$  and  $\hat{w}_i^{(t)} \rightarrow \hat{w}_i$ . Also, the properties of being  $v$ -lonely or  $v$ -popular with respect to  $t$  are increasing in  $t$ . A vertex  $v \in V$  is

lonely with respect to  $t$  if all its children are popular, so by Theorem 3.6, this occurs with probability  $(\hat{w}_2^{(t)} + o(1))^{\deg(v)}$ . The expected number of vertices in  $V$  that are lonely with respect to  $t$  is therefore  $f(\hat{w}_2^{(t)} + o(1))n = (f(\hat{w}_2) + o(1))n$ . To get concentration we use the second moment method. Let  $d(u, v)$  denote the distance between  $u$  and  $v$ . By coupling a random instance of the bipartite graph with an instance in which the  $t + 1$ -neighbourhood of  $v_2$  is fixed, it is clear that

$$|\mathbb{P}[v_1 \text{ lonely} \mid v_2 \text{ lonely}] - \mathbb{P}[v_1 \text{ lonely}]| \leq \mathbb{P}[d(v_1, v_2) < 2t + 1].$$

Hence, if  $X$  represents the number of lonely vertices in  $V$ ,

$$\begin{aligned} \text{Var}(X) &= \mathbb{E}(X^2) - \mathbb{E}(X)^2 \\ &= \sum_{v_1, v_2} (\mathbb{P}[v_1, v_2 \text{ lonely}] - \mathbb{P}[v_1 \text{ lonely}]\mathbb{P}[v_2 \text{ lonely}]) \\ &= n^2 O(\mathbb{P}[d(v_1, v_2) < 2t + 1]) = O(n\Delta^{2t}). \end{aligned}$$

Thus  $|X - \mathbb{E}(X)| \leq n^{1/2+\varepsilon}$  with high probability. Parts (b)-(d) are proved similarly. In (c) we note that as  $t \rightarrow \infty$  and the events of being popular and lonely are monotone, it is enough to count vertices  $v$  that have no  $v$ -lonely children and at least two children that are not popular. In (d) we are counting the number of these vertices with exactly  $k \geq 2$  children that are not  $v$ -popular. Statement (e) is an instant corollary of (d).  $\square$

### 3.5 Evolution of the generating functions

We generalise Theorem 3.7(d) by showing that after any number of steps of the Karp–Sipser algorithm, in either phase 1 or phase 2, the generating function of the degree distribution of the remaining vertices is of a simple form. Recall that we delete vertices that are used in the matching.

We shall use the differential equation method of Wormald (see [20]). However, to directly apply this method in the form proved in [20], we shall modify the Karp–Sipser algorithm slightly. Fix a  $\delta > 0$  and run the Karp–Sipser algorithm with the following modification. At each step, if the total number  $N_1$  of degree 1 vertices is less than  $\delta n$ , but more than zero, then with probability  $N_1/(\delta n)$  we run the original Karp–Sipser algorithm by picking uniformly at random an edge incident to a degree 1 vertex. However, with probability  $1 - N_1/(\delta n)$  we instead pick an edge uniformly from the graph (i.e., we don't notice that there are degree 1 vertices present). The advantage of this modification is that the derivatives in the differential equation method will now be Lipschitz. (In [4] a different

method we employed to avoid the discontinuity in the algorithm between  $N_1 = 0$  and  $N_1 > 0$ , namely steps were grouped together so that  $N_1 = 0$  always held after one of these multiple steps. Here however we are less concerned about the error terms so will use this simpler method.)

Let  $n_d(t)$  and  $\hat{n}_d(t)$  be the number of degree  $d$  vertices in  $V$  and  $\hat{V}$  respectively after  $t$  steps of the algorithm, and let  $\mathcal{F}_t$  be the filtration given by the edges revealed up to and including the  $t^{\text{th}}$  step. Write  $\Delta n_d(t) = n_d(t) - n_d(t-1)$  and define

$$\eta(t) = \frac{n_1(t)}{\max\{\delta n, n_1(t) + \hat{n}_1(t)\}}$$

and similarly for  $\hat{\eta}(t)$ . Then with probability  $\eta(t)$  the algorithm picks an edge adjacent to a degree 1 vertex in  $V$ , with probability  $\hat{\eta}(t)$  the algorithm picks an edge adjacent to a degree 1 vertex in  $\hat{V}$ , and with probability  $1 - \eta(t) - \hat{\eta}(t)$  an edge is picked uniformly at random from the entire graph. When an edge is chosen adjacent to a degree 1 vertex in  $\hat{V}$ , or arbitrarily in  $B$ , then the matching vertex in  $V$  is of degree  $d$  with probability  $dn_d(t)/E(t)$ , where  $E(t) = \sum dn_d(t) = \sum d\hat{n}_d(t)$  is the number of remaining edges in the graph. From this it is possible to calculate the expected change in the  $n_d(t)$  for  $0 \leq d \leq \Delta$  as

$$\begin{aligned} \mathbb{E}[\Delta n_d(t) \mid \mathcal{F}_{t-1}] &= -\eta\delta_{1d} - (1 - \eta)\frac{dn_d}{E} \\ &+ (1 - \hat{\eta})\left\{\frac{(d+1)n_{d+1} - dn_d}{E} \sum_{k=1}^{\Delta} \frac{k\hat{n}_k}{E}(k-1) + O(\Delta^3/E)\right\}. \end{aligned} \tag{3.7}$$

Here  $\delta_{1d} = 1$  if  $d = 1$  and  $\delta_{1d} = 0$  if  $d \neq 1$ . The first two terms gives the change caused in  $n_d(t)$  as a result of a degree  $d$  vertex in  $V$  being chosen in the matching. Either this vertex is chosen from one of the degree 1 vertices (with probability  $\eta$ ), or it is the vertex containing a randomly chosen configuration point. The last term gives the change in  $n_d$  caused by the removal of edges incident to the matching edge. If the matching edge meets a vertex of degree 1 in  $\hat{V}$  then no such edges meet  $V$  other than at the matching vertex. Otherwise we need to remove  $k-1$  configuration points from the  $V$  side where  $k$  is the degree of the matched vertex in  $\hat{V}$ . Removing one configuration point increases  $n_d$  by an average of  $\frac{1}{E}((d+1)n_{d+1} - dn_d)$ , i.e., the probability of this configuration point corresponds to a degree  $d+1$  vertex (which then becomes degree  $d$ ), minus the probability that this configuration point corresponds to a degree  $d$  vertex (which then becomes degree  $d-1$ ). If all  $k-1$  removed configuration



points corresponded to distinct vertices, then we could just multiply  $\frac{1}{E}((d+1)n_{d+1} - dn_d)$  by  $k-1$ . There is however an error when  $k > 2$  as it is possible for the degree of a vertex to drop by more than 1 when there are multiple edges in  $B$ . We bound the expected contribution of this by the number of pairs of configuration points in  $\hat{V}$ , namely  $\binom{k-1}{2} = O(\Delta^2)$ , multiplied by the probability that they match to the same vertex in  $V$ , which is  $O(\Delta/E)$ .

We can rewrite (3.7) in the form

$$\mathbb{E}[\Delta n_d(t) \mid \mathcal{F}_{t-1}] = h_d\left(\frac{n_1(t)}{n}, \dots, \frac{n_\Delta(t)}{n}, \frac{\hat{n}_1(t)}{n}, \dots, \frac{\hat{n}_\Delta(t)}{n}\right) + O(1/n)$$

with

$$\begin{aligned} h_d(\zeta, \hat{\zeta}) &= -\delta_{d1}\eta - (1-\eta)\phi^{-1}d\zeta_d \\ &\quad + (1-\hat{\eta})\phi^{-2}((d+1)\zeta_{d+1} - d\zeta_d) \sum (k-1)k\hat{\zeta}_k, \end{aligned} \quad (3.8)$$

where  $\zeta = (\zeta_0, \dots, \zeta_\Delta)$ ,  $\hat{\zeta} = (\hat{\zeta}_0, \dots, \hat{\zeta}_\Delta)$ ,  $\phi = \sum d\zeta_d = \sum d\hat{\zeta}_d$ ,  $\eta = \zeta_1 / \max\{\delta, \zeta_1 + \hat{\zeta}_1\}$ , and  $\hat{\eta} = \hat{\zeta}_1 / \max\{\delta, \zeta_1 + \hat{\zeta}_1\}$ . We define  $\hat{h}_d$  similarly and note that  $h_d$  and  $\hat{h}_d$  are Lipschitz on the domain

$$D = \{(\zeta, \hat{\zeta}) \in [0, \infty)^{2\Delta+2} : \phi \geq \sqrt{\delta}\}.$$

Hence we can apply Theorem 5.1 of [20] (with  $\gamma = 0$ ,  $\beta = \Delta$ ,  $\lambda = n^{-1/4}$ ) to deduce that

$$n_d(t) = n\zeta_d(t) + O(n^{3/4}), \quad \hat{n}_d(t) = n\hat{\zeta}_d(t) + O(n^{3/4}),$$

with probability  $1 - e^{-\Omega(n^{1/4})}$  uniformly in the set of  $t$  for which  $\sum d\zeta_d(t) \geq \sqrt{\delta}$ , where  $\zeta_d(t)$  and  $\hat{\zeta}_d(t)$  are the (unique) solutions to the system of differential equations

$$\frac{d}{dt}\zeta_d(t) = h_d(\zeta(t), \hat{\zeta}(t)), \quad \frac{d}{dt}\hat{\zeta}_d(t) = \hat{h}_d(\zeta(t), \hat{\zeta}(t)), \quad d = 0, \dots, \Delta,$$

with initial conditions  $\zeta_d(0) = z_d$ ,  $\hat{\zeta}_d(0) = \hat{z}_d$ .

**Lemma 3.8** *Running the modified Karp-Sipser's algorithm yields with high probability at any stage a bipartite graph such that there exist  $\alpha, \beta, \gamma, c$  and  $\hat{\alpha}, \hat{\beta}, \hat{\gamma}, \hat{c}$  so that the degree generating functions of the bipartite graph are  $f(\alpha + \beta x) - \gamma x - c + \varepsilon(x)$  and  $\hat{f}(\hat{\alpha} + \hat{\beta} x) - \hat{\gamma} x - \hat{c} + \hat{\varepsilon}(x)$ , where  $\varepsilon(x) = \sum_{i=2}^{\Delta} \varepsilon_i x^i$ ,  $\hat{\varepsilon}(x) = \sum_{i=2}^{\Delta} \hat{\varepsilon}_i x^i$  with  $\varepsilon_i, \hat{\varepsilon}_i = o(1)$ . Moreover, the coefficients*

$\alpha, \beta, \gamma, \hat{\alpha}, \hat{\beta}, \hat{\gamma}$  evolve according to the coupled differential equations

$$\left. \begin{aligned} \frac{d}{dt}\alpha &= \beta\hat{\beta}^2(1-\hat{\eta})\phi^{-2}\hat{f}''(\hat{\alpha}+\hat{\beta}), \\ \frac{d}{dt}(\alpha+\beta) &= -(1-\eta)\phi^{-1}, \\ \frac{d}{dt}\gamma &= -\eta - \frac{\gamma}{\beta}\frac{d\beta}{dt}, \end{aligned} \right\} \quad (3.9)$$

where

$$\eta = \zeta_1 / \max\{\delta, \zeta_1 + \hat{\zeta}_1\}, \quad \zeta_1 = \beta f'(\alpha) - \gamma,$$

$$\phi = \beta f'(\alpha + \beta) - \gamma = \hat{\beta} \hat{f}'(\hat{\alpha} + \hat{\beta}) - \hat{\gamma}$$

and similarly for  $\hat{\alpha}, \hat{\beta}, \hat{\gamma}, \hat{\eta}, \hat{\zeta}_1$ .

**Proof** Write  $f_t(x) = \sum_{d=0}^{\Delta} \zeta_d(t)x^d$ ,  $\hat{f}_t(x) = \sum_{d=0}^{\Delta} \hat{\zeta}_d(t)x^d$ . Now from (3.8),

$$\frac{d}{dt}f_t(x) = -\eta x - (1-\eta)\frac{1}{f'_t(1)}xf'_t(x) + (1-\hat{\eta})\frac{\hat{f}''_t(1)}{\hat{f}'_t(1)^2}(1-x)f'_t(x).$$

as  $\phi = f'_t(1) = \hat{f}'_t(1)$ . Now this is of the form  $\frac{d}{dt}f_t(x) = -A(t)x - B(t)xf'_t + C(t)(1-x)f'_t$ , whose solution is easily seen to be of the form

$$f_t(x) = f(\alpha + \beta x) - \gamma x - c$$

for some functions  $\alpha(t)$ ,  $\beta(t)$ ,  $\gamma(t)$ , and  $c(t)$  satisfying (3.9).  $\square$

**Proof** [Proof of Theorem 3.1] We run the 1st phase of Karp–Sipser for  $t_0$  rounds,  $t_0 = o(\log n)$ ,  $t_0 \rightarrow \infty$ . After this phase the generating function for the degrees in  $V$  is given approximately by  $f_{t_0}(x)$  which by Theorem 3.8 is of the form  $f(\alpha + \beta x) - f'(\alpha)x - c$  for some constant  $c$ . Here  $\alpha = \hat{w}_2$ ,  $\beta = 1 - \hat{w}_1 - \hat{w}_2$ . The number of vertices in  $V$  that become isolated up to time  $t$  is by Lemma 3.5 the difference between the number of lonely vertices in  $V$  and the number of popular vertices in  $\hat{V}$ , which by Theorem 3.7(b) is just

$$(f(\hat{w}_2) - \hat{f}(1) + \hat{f}(1 - w_1) + \hat{f}'(1 - w_1)w_1)n + o(n).$$

By Theorem 3.7(c) there are  $(f(1 - \hat{w}_1) - f(\hat{w}_2) - f'(\hat{w}_2)\hat{w}_2)n + o(n)$  vertices with degree at least 2, and if all, or almost all, of these are matched then the size of the matching will be

$$\{f(1) - (f(\hat{w}_2) - \hat{f}(1) + \hat{f}(1 - w_1) + \hat{f}'(1 - w_1)w_1)\}n + o(n) = (\xi - o(1))n.$$

Assume that  $\xi \leq \hat{\xi}$ , so that after phase 1 there are at least as many non-isolated vertices in  $\hat{V}$  as in  $V$ . We then run phase 2 and bound the

number of vertices of  $V$  that become isolated. We note that if throughout the evolution of the process  $\zeta_1(t) \leq \delta$ , then the number of vertices that become isolated is small. Indeed, by (3.8),

$$\frac{d}{dt}\zeta_0 = h_0(\zeta(t), \hat{\zeta}(t)) = (1 - \hat{\eta})\phi^{-2}\zeta_1 \sum (k-1)k\hat{\zeta}_k$$

and  $\sum (k-1)k\hat{\zeta}_k \leq \Delta\phi$ , so  $\frac{d}{dt}\zeta_0 = O(\zeta_1/\phi) = O(\sqrt{\delta})$  when  $\phi \geq \sqrt{\delta}$ . Thus  $\zeta_0(t) = O(n\sqrt{\delta})$  for all relevant  $t$ , and hence is  $o(n)$  as  $\delta \rightarrow 0$ .

Thus it is enough to show that if  $\zeta_1 \geq \delta$  then  $\frac{d\zeta_1}{dt} \leq 0$ , or equivalently  $h_1(\zeta, \hat{\zeta}) \leq 0$ . From (3.8), this is equivalent to

$$-\eta - (1 - \eta)\phi^{-1}\zeta_1 + (1 - \hat{\eta})\phi^{-2}(2\zeta_2 - \zeta_1) \sum (k-1)k\hat{\zeta}_k \leq 0.$$

However,  $\zeta_1 \geq \delta$  implies that  $1 - \hat{\eta} = \eta$ , so it is enough if

$$\phi^{-2}2\zeta_2 \sum (k-1)k\hat{\zeta}_k \leq 1.$$

Now  $\phi = f'_t(1) = \hat{f}'_t(1)$ ,  $\sum (k-1)k\hat{\zeta}_k = f''_t(1)$ , and  $2\zeta_2 = f''_t(0)$ . Thus it is enough that

$$\frac{f''_t(0)}{f'_t(1)} \cdot \frac{\hat{f}''_t(1)}{\hat{f}'_t(1)} \leq 1 \quad (3.10)$$

for all  $t$ . Now  $f''_t(0) = \beta^2 f''(\alpha)$ ,  $\hat{f}''_t(1) = \hat{\beta}^2 \hat{f}''(\hat{\alpha} + \hat{\beta})$ , and  $f'_t(1) \geq \beta(f'(\alpha + \beta) - f'(\alpha))$ ,  $\hat{f}'_t(1) \geq \hat{\beta}(\hat{f}'(\hat{\alpha} + \hat{\beta}) - \hat{f}'(\hat{\alpha}))$ . Thus (3.10) is implied by (3.4). Note that it is enough to require this only along the trajectory of  $(\alpha, \beta, \hat{\alpha}, \hat{\beta})$  taken by the algorithm in phase 2. This trajectory is given by Lemma 3.8.  $\square$

**Proof** [Proof of Theorem 3.2.] In the symmetric case we have  $\hat{f} = f$ ,  $\hat{\alpha} = \alpha$ ,  $\hat{\beta} = \beta$ . Thus it is enough to check (3.4) for all  $0 \leq \alpha < \alpha + \beta \leq 1$  as this covers all possible trajectories for any  $\delta > 0$ . In the symmetric case this is equivalent to

$$f''(\alpha)f''(\alpha + \beta) \leq ((f'(\alpha + \beta) - f'(\alpha))/\beta)^2. \quad (3.11)$$

The result then follows from the following lemma.  $\square$

**Lemma 3.9** *Equation (3.11) holds for all  $0 \leq \alpha < \alpha + \beta \leq 1$  if and only if  $(f''(x))^{-1/2}$  is convex in  $[0, 1]$ .*

**Proof** First we note that, as  $f(x)$  is a polynomial and hence infinitely differentiable, the convexity of  $f''(x)^{-1/2}$  is equivalent to the condition

$$2f''(x)f^{(4)}(x) \leq 3f^{(3)}(x)^2. \quad (3.12)$$

Indeed

$$\begin{aligned} \frac{d^2}{dx^2} f''(x)^{-1/2} &= -\frac{1}{2} \frac{d}{dx} f''(x)^{-3/2} f^{(3)}(x) \\ &= \frac{1}{4} f''(x)^{-5/2} (3f^{(3)}(x)^2 - 2f''(x)f^{(4)}(x)). \end{aligned}$$

Now suppose (3.11) holds for all  $0 \leq \alpha < \alpha + \beta \leq 1$ . Expanding both sides of (3.11) as a Taylor series up to  $\beta^2$  gives

$$\begin{aligned} f''(\alpha)(f''(\alpha) + \beta f^{(3)}(\alpha) + \frac{\beta^2}{2} f^{(4)}(\alpha) + O(\beta^3)) \\ \leq (f''(\alpha) + \frac{\beta}{2} f^{(3)}(\alpha) + \frac{\beta^2}{6} f^{(4)}(\alpha) + O(\beta^3))^2, \end{aligned}$$

which after some simplification gives

$$2f''(\alpha)f^{(4)}(\alpha) \leq 3f^{(3)}(\alpha)^2 + O(\beta). \quad (3.13)$$

Letting  $\beta \rightarrow 0$  gives (3.12) for all  $x = \alpha \in [0, 1)$  (and hence also at  $x = 1$  by continuity) as required.

Now assume  $f''(x)^{-1/2}$  is convex. Then for  $t \in [0, 1]$ ,

$$f''(\alpha + t\beta)^{-1/2} \leq c_0(1 - t) + c_1 t,$$

where  $c_0 = f''(\alpha)^{-1/2}$  and  $c_1 = f''(\alpha + \beta)^{-1/2}$ . Thus

$$\begin{aligned} f'(\alpha + \beta) - f'(\alpha) &= \int_{\alpha}^{\alpha+\beta} f''(x) dx \\ &\geq \beta \int_0^1 (c_0(1 - t) + c_1 t)^{-2} dt \\ &= \beta \left[ - (c_1 - c_0)^{-1} (c_0(1 - t) + c_1 t)^{-1} \right]_0^1 \\ &= \frac{\beta}{c_1 - c_0} \left( \frac{1}{c_0} - \frac{1}{c_1} \right) \\ &= \frac{\beta}{c_0 c_1} = \beta (f''(\alpha) f''(\alpha + \beta))^{1/2} \end{aligned}$$

which is just (3.11). □

The proof of Theorem 3.3 is exactly analogous, so we omit it. Finally we show that the condition used in [4], namely the log-concavity of the sequence  $dz_d$ , is enough to deduce the convexity of  $f''(x)^{-1/2}$  and hence apply Theorem 3.3.

**Lemma 3.10** *If the sequence  $dz_d$  is log-concave then  $f''(x)^{-1/2}$  is convex on  $[0, 1]$ .*

**Proof** Let  $g(x) := f'(x) = \sum_{d=1}^{\Delta} dz_d x^{d-1}$ . If we write  $g(x) = \sum a_n x^n$ , then log-concavity of  $(dz_d)$  is equivalent to the log-concavity of  $(a_n)$ . Write  $g(x+h) = \sum a_n(x)h^n$ , so that  $a_n(x) = \frac{1}{n!} \frac{d^n}{dx^n} g(x)$  and  $a_n(0) = a_n$ . We first show that for all  $x \geq 0$ ,  $(a_n(x))$  is log-concave. Log-concavity of  $(a_n(x))$  at one value of  $x$  is equivalent to  $\varepsilon_{n,k}(x) := a_{n+k-1}(x)a_{n+1}(x) - a_{n+k}(x)a_n(x) \geq 0$  for all  $k \geq 2$ ,  $n \geq 0$ , see for example [13]. Now  $da_n(x)/dx = (n+1)a_{n+1}(x)$ . Hence

$$\begin{aligned} \frac{d}{dx} \varepsilon_{n,k}(x) &= (n+k)a_{n+k}(x)a_{n+1}(x) + (n+2)a_{n+k-1}(x)a_{n+2}(x) \\ &\quad - (n+k+1)a_{n+k+1}(x)a_{n+1}(x) - (n+1)a_{n+k}(x)a_{n+1}(x) \\ &= (k-1)a_{n+k}(x)a_{n+1}(x) + (n+2)a_{n+k-1}(x)a_{n+2}(x) \\ &\quad - (n+k+1)a_{n+k+1}(x)a_{n+1}(x) \\ &= (k-1)\varepsilon_{n+1,k}(x) + (n+2)\varepsilon_{n+2,k-1}(x) \end{aligned}$$

As  $\varepsilon_{n,1}(x) = 0$ , and  $\varepsilon_{n,k}(0) \geq 0$  for  $k \geq 2$ , we deduce that  $\varepsilon_{n,k}(x) \geq 0$  for all  $x \geq 0$  and  $k \geq 2$ .

Now as  $(a_n(x))$  is log-concave,  $a_1(x)a_3(x) \leq a_2(x)^2$ . It follows that  $\frac{1}{6}g'(x)g^{(3)}(x) \leq \frac{1}{4}g''(x)^2$ , or equivalently  $2f''(x)f^{(4)}(x) \leq 3f^{(3)}(x)^2$ , as required.  $\square$

## 4 Matchings with an adversary

In this section we are interested in the problem of finding the largest matching in a random bipartite graph after an adversary has deleted some edges. More precisely, we consider a random bipartite graph  $G = (A \cup B, E)$  with  $|A| = n$  and  $|B| = (1 + \varepsilon)n$ . Each vertex in  $A$  is adjacent to  $d$  neighbours chosen uniformly at random from  $B$ . We allow repetition, so this is a multigraph. An adversary is then able to remove a single edge adjacent to each vertex of  $A$ , with the aim of minimising the size of the largest matching. We want find to the smallest  $\varepsilon$  such that a matching of size  $n$  exists. Note that this problem corresponds to finding a small control set in a directed network where each node has  $d$ -out-neighbours and an adversary can manipulate one out-arc at each node. Our result

then implies that we can control roughly  $(1 - \log d/d^2)n$  nodes with a single node, which implies that there is a control set of size about  $\log d/d^2$ .

**Theorem 4.1** [2] *Let  $G = (A \cup B, E)$  with  $|A| = n$  and  $|B| = (1 + \varepsilon)n$  be a random bipartite (multi-)graph in which each vertex in  $A$  chooses  $d$  vertices uniformly at random with repetition from  $B$ . For each  $\eta > 0$  there exists a  $d_0$  such that for all  $d \geq d_0$ , if  $\varepsilon > (4 + \eta) \log d/d^2$ , then with high probability an adversary who deletes one edge incident to each vertex in  $A$  cannot destroy all matchings of size  $n$ . On the other hand if  $\varepsilon < (4 - \eta) \log d/d^2$ , then asymptotically almost surely such an adversary can destroy all matchings of size  $n$ .*

The problem of finding such resilient matchings is closely related to finding a maximum strongly independent set in  $d$ -uniform hypergraphs  $\mathcal{H}_{\tilde{n},n}^d$  on  $\tilde{n} \geq n$  vertices and  $n$  edges. A strongly independent set in a  $d$ -uniform hypergraph is a set of vertices such that no hyperedge contains two or more vertices of this set, see for example [3]. Here, the partition class  $B$  of the bipartite graph  $G$  corresponds to the vertices of the hypergraph and the neighbourhoods of the vertices of  $A$  correspond to the edges (multi-edges are a small nuisance but one can show with Markov's inequality that whp there are not many, say less than  $\sqrt{\tilde{n}}$ ). Clearly, an adversary can isolate all vertices corresponding to a strongly independent set, and hence if the maximum size of a strongly independent set is  $\beta$  then the adversary can force the size of a maximum matching to be at most  $\tilde{n} - \beta$ . We will show that  $\mathcal{H}_{\tilde{n},n}^d$  contains whp a strongly independent set of size  $(4 - \eta)n \log d/d^2$  where  $\eta$  can be chosen arbitrarily small if  $d$  is sufficiently large.

As a note on notation, asymptotically, we are interested in the probabilistic results as  $n \rightarrow \infty$ , and so by  $o(1)$  we mean a function that tends to 0 as  $n \rightarrow \infty$ , but since we are also considering  $d \rightarrow \infty$  (and at the same time  $\varepsilon \rightarrow 0$ ), we may also require the use of little  $o$  notation to denote the size of terms which do not depend on  $n$ , in which case we will label them  $o_d(f(d))$  to indicate that the asymptotics depend on  $d$  rather than  $n$ . As an example  $\varepsilon = o_d(1)$ , but neither depend on  $n$  and hence are asymptotically constant in terms of  $n$ .

To prove that there exists a matching of size  $n = |A|$  if  $|B|$  is sufficiently large we use Hall's theorem [10]. Recall that by Hall's theorem for a bipartite graphs  $G$  with partition classes  $X$  and  $Y$ , a matching of size  $|X|$  exists if and only if, for all  $X' \subseteq X$ ,  $|\Gamma(X')| \geq |X'|$ . Here  $\Gamma(X')$  is the set of neighbours of  $X'$  (in  $Y$ ). With a careful (and lengthy) analysis we show in [2] that Hall's condition is satisfied and obtain the following theorem.

**Theorem 4.2** *Let  $G$  be the random bipartite graph with partition sets  $A$  and  $B$  of size  $n$  and  $\tilde{n} = (1+\varepsilon)n$  respectively, and each vertex of  $A$  chooses  $d$  vertices uniformly at random with repetition from  $B$ . An adversary deletes a single edge incident to each vertex of  $A$  to obtain  $G'$ . For each  $\eta > 0$  there exists a  $d_0$  such that for all  $d \geq d_0$  and  $\varepsilon > (4 + \eta)\log d/d^2$ , a matching of size  $n$  still exists in  $G'$  with probability tending to 1 as  $n \rightarrow \infty$ .*

To prove that there is no matching of size  $n = |A|$  if  $|B|$  is too small we consider strongly independent sets in random  $d$ -uniform hypergraphs as discussed above.

**Theorem 4.3** *For each  $\eta > 0$  there exists a  $d_0$  such that for all  $d \geq d_0$ , there exists a strongly independent set in the random  $d$ -uniform hypergraph  $\mathcal{H}_{\tilde{n},n}^d$  consisting of  $\tilde{n} \geq n$  vertices and  $n$  edges, which asymptotically almost surely is at least of size*

$$(4 - \eta) \frac{\log d}{d^2} n.$$

The proof uses an approach by Frieze [8] suggested by Luczak. We use a partition  $P$  of the vertex set consisting of parts  $P_1, \dots, P_{n'}$  of size  $m$  where  $m$  grows asymptotically faster than  $(\log d)^2$  but slower than  $d^2/\log d$ , that is,  $m = o_d(d^2/\log d)$  and  $m = \omega_d((\log d)^2)$ . A set is a  $P$ -set if it is strongly independent and contains at most one vertex from each  $P_i$  for  $i \in \{1, \dots, n'\}$ . Let  $\beta$  be the maximum size of a  $P$ -set. By Azuma's inequality we have that

$$\mathbb{P}(|\beta - \mathbb{E}(\beta)| \geq t) \leq 2e^{-t^2/2n'} \quad (4.1)$$

using the martingale that exposes the hyperedges incident to  $P_i$  at step  $i$ ,  $i = 1, \dots, n'$ , and noting that  $\beta$  can change by at most 1 if we change hyperedges incident to a single  $P_i$ .

Let  $X_k$  be the random variable that counts the number of  $P$ -sets of size  $k$ . Using the second moment method we show that for  $\eta > 0$  and sufficiently large  $d$ ,

$$\mathbb{P}(X_k > 0) > 2 \exp \left( -2^9 \left( \frac{\log d}{d} \right)^4 n \right) \quad (4.2)$$

where

$$k = (4 - \psi) \frac{\log d}{d^2} n,$$

and  $\eta = 2\psi$ .

This implies the lower bound by setting  $t = 2^5 \left(\frac{\log d}{d}\right)^2 n/\sqrt{m}$  as

$$t^2/2n' = \frac{2^{10}n^2}{2n'm} \left(\frac{\log d}{d}\right)^4 = \frac{2^{10}n^2}{2n} \left(\frac{\log d}{d}\right)^4 = 2^9 \left(\frac{\log d}{d}\right)^4 n.$$

From this it follows that the probability that  $X_k > 0$  is larger than the probability that  $\beta$ , (which is the largest  $k$  for which  $X_k \neq 0$ ) lies further from the expectation than  $t$ . Thus  $k$  must be at most  $t$  greater than  $\mathbb{E}(\beta)$ . By Azuma's inequality we know that with high probability  $|\beta - \mathbb{E}(\beta)| < t$  and hence with high probability  $\beta > k - 2t$ . Hence if  $m/(\log d)^2 \rightarrow \infty$ , we have that for sufficiently large  $d$ , whp

$$\beta > (4 - \psi - 2^7(\log d)/\sqrt{m}) \frac{\log d}{d^2} n \geq (4 - 2\psi) \frac{\log d}{d^2} n = (4 - \eta) \frac{\log d}{d^2} n.$$

#### 4.1 Conclusions/Open problems

Although we have proven a threshold for the lower and upper bounds which are asymptotically equal as  $d \rightarrow \infty$ , it seems likely, that a threshold should exist for each  $d$ . In other words, we conjecture that there exists constants  $c_d$  for  $d \geq 3$  such that for all  $\eta > 0$ , if  $\varepsilon > c_d + \eta$  then whp a matching of size  $n$  can be found, while for  $\varepsilon < c_d - \eta$  there is whp a strategy for the adversary that reduces the size of the maximal matching below  $n$ . If this conjecture is true then we know  $c_d = (4 + o_d(1))(\log d)/d^2$ . Note that this conjecture fails for  $d = 2$ . Indeed, the adversary can simply delete a random choice of edge from each vertex in  $A$  and then whp in the resulting graph we have two vertices in  $A$  with the same remaining neighbour in  $B$ . On the other hand it is not hard to see that Theorem 4.2 can be strengthened so as to give a finite bound on  $\varepsilon$  even for  $d = 3$ . The proof required  $d$  to be large at several points, however, following the strategy of the proof, it is easy to show that  $d \geq 3$  is enough to get a finite bound. For example, if  $\tilde{n} \geq 10dn$  then, recalling that  $e_s$  is the expected number of witnesses of size  $s$  and  $p := \frac{s-1}{n}$ , we have,

$$\begin{aligned} e_s &\leq \binom{n}{s} \binom{\tilde{n}}{s-1} ((qd + p)p^{d-1})^s \\ &\leq (en/s)^s (e\tilde{n}/(s-1))^{s-1} (dp^2)^s \\ &\leq (e/10dp)^s (e/p)^{s-1} (dp^2)^s \\ &\leq (p/e)(e^2/10)^s \\ &\leq (s/ne)(e^2/10)^s. \end{aligned}$$



Noting that

$$\sum_{s=1}^n (s/ne)(e^2/10)^s = O\left(\frac{1}{n}\right),$$

it is clear that we have  $e_s = o(1)$  as required.

Although we have identified the threshold for which an adversary can and cannot destroy the complete matching when subject to the restriction of removing a single edge incident to each vertex of  $A$ , there remains a number of interesting problems that would arise from allowing the adversary greater or differing powers in modifying  $G$ . The case where the adversary is able to delete  $n$  edges globally allows the adversary to easily isolate a linear proportion of the vertices, while equally, a matching still exists that covers a linear proportion of the vertices. In both cases a simple greedy algorithm provides fairly simple bounds, but finding the exact size of the largest remaining matching seems challenging and certainly would require further insight in tackling.

Another problem to consider would be the case of analysing the size of the maximum matching for values of  $d$  and  $\varepsilon$  for which we have shown that the adversary can eliminate a matching of size  $n$ . The use of our graph model was motivated by its use in [9] which analysed the size of the maximum matching in the same model but without an adversary removing edges, for all values of  $d$ , and it would be interesting to know what the behaviour of the size of the maximum matching becomes for small values of  $d$  once an adversary is introduced.

**Acknowledgement:** The first author was partially supported by NSF grant DMS 1301614. We would also like to thank the referee for the careful reading of the manuscript and the helpful comments.

## References

- [1] J. Aronson, A. Frieze and B. Pittel, *Maximum matchings in sparse random graphs: Karp–Sipser revisited*, Random Structures & Algorithms **12** (1998), 111–177.
- [2] P.N. Balister, S. Gerke and A. McDowell, *Adversarial resilience of matchings in bipartite random graphs*, manuscript.
- [3] C. Berge, *Hypergraphs*, volume 45 of North-Holland Mathematical Library, North-Holland Publishing Co., Amsterdam, 1989.
- [4] T. Bohman, A.M. Frieze, *Karp–Sipser on random graphs with a fixed degree sequence*, Combin. Probab. Comput. **20** (2011), 721–741.

- [5] C. Bordenave, M. Lelarge, and J. Salez, *Matchings on infinite graphs*, Probab. Theory Related Fields **157** (2013), no. 1–2, 183–208.
- [6] J.-M. Dion, C. Commault, and J. van der Woude, *Generic properties and control of linear structured systems*, Automatica J. IFAC, **39** (2003), no. 7, 1125–1144.
- [7] J. Edmonds, *Paths, trees and flowers*, Canad. J. Math. **17** (1965), 449–467.
- [8] A.M. Frieze, *On the independence number of random graphs*, Discrete Math. **81** (1990), no. 2, 171–175.
- [9] A. Frieze and P. Melsted, *Maximum matchings in random bipartite graphs and the space utilization of Cuckoo Hash tables*, Random Structures & Algorithms **41** (2012), no. 3, 334–364.
- [10] P. Hall, *On representatives of subsets*, J. Lond. Math. Soc. **s1-10** (1935), no. 1, 26–30.
- [11] S. Janson, T. Łuczak, A. Ruciński, *Random graphs*, Wiley-Intersci. Ser. Discrete Math. Optim. Wiley-Interscience, New York (2000).
- [12] R.E. Kalman, *Mathematical description of linear dynamical systems*, J. SIAM Ser. A Control **1** (1963) 152–192.
- [13] B. Sagan, *Log-concave sequences of symmetric functions and analogs of the Jacobi-Trudi determinants*, Trans. Amer. Math. Soc. **329** (1992) 795–811.
- [14] R. Karp, M. Sipser, *Maximum matchings in sparse random graphs*, Proceedings of the 22nd IEEE Symposium on the Foundations of Computer Science (1981) 364–375.
- [15] C.-T. Lin, *Structural controllability*, IEEE Trans. Automatic Control **19** (1974) 201–208.
- [16] Y.-Y. Liu, J.-J. Slotine, A.-L. Barabási, *Controllability of complex networks*, Nature **473** (2011) 167–173.
- [17] S. Micali and V. Vazirani, *An  $O(|V|^{\frac{1}{2}}|E|)$  algorithm for finding maximum matchings in general graphs*, Proceedings of the 21st IEEE Symposium on the Foundations of Computer Science (1980) 17–27.
- [18] J. Salez, *Weighted enumeration of spanning subgraphs in locally tree-like graphs*, Random Structures & Algorithms **43** (2013), no. 3, 377–397.

- [19] R.W. Shields and J.B. Pearson, *Structural controllability of multi-input linear systems*, IEEE Trans. Automatic Control **21** (1976), 203–212.
- [20] N.C. Wormald, *The differential equation method for random graph processes and greedy algorithms*, In M. Karoński and H.J. Prömel, editors, Lectures on approximation and randomized algorithms, Warsaw (1999) 75–152.

University of Memphis  
Department of Math Sciences  
Memphis, TN 38152, USA  
pbalistr@memphis.edu

Royal Holloway University of London  
Mathematics Department  
Egham TW20 0EX, UK  
stefanie.gerke@rhul.ac.uk