

# Finding perfect matchings in random cubic graphs in linear expected time

Michael Anastos and Alan Frieze\*

November 12, 2018

## Abstract

In a seminal paper on finding large matchings in sparse random graphs, Karp and Sipser [12] proposed two algorithms for this task. The second algorithm has been intensely studied, but due to technical difficulties, the first algorithm has received less attention. Empirical results in [12] suggest that the first algorithm is superior. In this paper we show that this is indeed the case, at least for random cubic graphs. We show that w.h.p. the first algorithm will find a matching of size  $n/2 - O(\log n)$  on a random cubic graph (indeed on a random graph with degrees in  $\{3, 4\}$ ). We also show that the algorithm can be adapted to find a perfect matching w.h.p. in  $O(n)$  time, as opposed to  $O(n^{3/2})$  time for the worst-case.

## 1 Introduction

Given a graph  $G = (V, E)$ , a matching  $M$  of  $G$  is a subset of edges such that no vertex is incident to two edges in  $M$ . Finding a maximum cardinality matching is a central problem in algorithmic graph theory. The most efficient algorithm for general graphs is that given by Micali and Vazirani [13] and runs in  $O(|E||V|^{1/2})$  time.

In a seminal paper, Karp and Sipser [12] introduced two simple greedy algorithms for finding a large matching in the random graph  $G_{n,m}$ ,  $m = cn/2$  for some positive constant  $c > 0$ . Let us call them Algorithms 1 and 2 as they are in [12]. Algorithm 2 is simpler than Algorithm 1 and has been intensely studied: see for example Aronson, Frieze and Pittel [1], Bohman and Frieze [3], Balister and Gerke [2] or Bordenave and Lelarge [6]. In particular, [1] together with Frieze and Pittel [9] shows that w.h.p. Algorithm 2 finds a matching that is within  $\tilde{O}(n^{1/5})$  of the optimum, when applied to  $G_{n,m}$ . Subsequently, Chebolu, Frieze and Melsted [5] showed how to use Algorithm 2 as a basis for a linear expected time algorithm, when  $c$  is sufficiently large.

---

\*Research supported in part by NSF Grant DMS1363136

Algorithm 2 proceeds as follows (a formal definition of Algorithm 1 is given in the next section). While there are isolated vertices it deletes them. After which, while there are vertices of degree one in the graph, it chooses one at random and adds the edge incident with it to the matching and deletes the endpoints of the edge. Otherwise, if the current graph has minimum degree at least two, then it adds a random edge to the matching and deletes the endpoints of the edge.

In the same paper Karp and Sipser proposed another algorithm for finding a matching that also runs in linear time. This was Algorithm 1. The algorithm sequentially reduces the graph until it reaches the empty graph. Then it unwinds some of the actions that it has taken and grows a matching which is then output. Even though it was shown empirically to outperform Algorithm 2, it has not been rigorously analyzed. In this paper we analyze Algorithm 2 in the special case where the graph is random with a fixed degree sequence  $3 \leq d(i) \leq 4$  for  $i = 1, 2, \dots, n$ . We prove the following:

**Theorem 1.** *Let  $G$  be a random graph with degree sequence  $3 \leq d(i) \leq 4$  for  $i = 1, 2, \dots, n$ . Then*

- (a) *Algorithm 1 finds a matching of size  $n/2 - O(\log n)$ , w.h.p.*
- (b) *Algorithm 1 can be modified to find a (near) perfect matching in  $O(n)$  time w.h.p. and in expectation.*

A (near) perfect matching is one of size  $\lfloor n/2 \rfloor$ . Note that in the case of cubic graphs, it is known that they have (near) perfect matchings w.h.p., see Bollobás [4]. Note also that it was shown by Frieze, Radcliffe and Suen [10] that w.h.p. Algorithm 2 finds a matching of size  $n/2 - \tilde{O}(n^{1/5})$ .<sup>1</sup>

## 2 The Algorithm

The algorithm that is given in [12] can be split into two parts. The first part sequentially reduces the graph until it reaches the empty graph. Then the second part reverses part of this reduction and grows a matching which is then output.

To reduce the graph,

- (1) First, while there are vertices of degree 0 or degree 1 the algorithm removes them along with any edge incident to them. The edges removed at this stage will be part of the output matching.
- (2) Second, while there are vertices of degree 2 the algorithm contracts them along with their two neighbors. That is the induced path  $(x, y, z)$  is replaced by a single contracted vertex  $y_c$  whose neighbors are those of  $x, z$  other than  $y$ . The description in [12] does

---

<sup>1</sup>Recently, the junior author has extended Theorem 1 to random  $r$ -regular graphs for all  $3 \leq r = O(1)$ .

not explicitly say what to do with loops or multiple edges created by this process. In any case, such creations are very rare. We say a little more on this in Section 2.2.

In the unwinding, if we have so far constructed a matching containing an edge  $\{y_c, \xi\}$  incident with  $y_c$  and  $\xi$  is a neighbor of  $x$  then in our matching we replace this edge by  $\{x, \xi\}$  and  $\{y, z\}$ . If there is no matching edge so far chosen incident with  $y_c$  then we add an arbitrary one of  $\{x, y\}$  or  $\{y, z\}$  to our matching.

- (3) Finally if the graph has minimum degree 3 then a random vertex is chosen among those of maximum degree and then a random edge incident to that vertex is deleted. These edges will not be used in the unwinding.

## 2.1 Idea of proof:

No mistakes are made while handling vertices of degree 0,1 or 2. Each mistake decreases the size of the final matching produced by one from the maximum size. We will show that mistakes occur only at parts of the graph that have become denser than is likely.

We show that w.h.p. the maximum degree remains  $O(\log^2 \nu)$  where  $\nu$  is the number of vertices remaining and so as long as  $\nu \log n$ , say, then w.h.p. there will be no dense subgraphs and the algorithm will not make any mistakes. This explains the  $O(\log n)$  error term. Finally, we assert that removing an edge incident to a vertex of a maximum degree will help to control the maximum degree, explaining this choice of edge to delete.

## 2.2 Details

The precise algorithm that we analyze is called REDUCE-CONSTRUCT The algorithm description given in [12] is not explicit in how to deal with loops and multiple edges, as they arise. We will remove loops immediately, but keep the multiple edges until removed by other operations.

We assume that our input (multi-)graph  $G = G([n], E)$  has degree sequence  $\mathbf{d}$  and is generated by the configuration model of Bollobás [4]. Let  $W = [2\nu]$ ,  $2\nu = \sum_{i=1}^n d(i)$ , be our set of *configuration points* and let  $\Phi$  be the set of *configurations* i.e. functions  $\phi : W \mapsto [n]$  that such that  $|\phi^{-1}(i)| = d(i)$  for every  $i \in [n]$ . Given  $\phi \in \Phi$  we define the graph  $G_\phi = ([n], E_\phi)$  where  $E_\phi = \{\{\phi(2j-1), \phi(2j)\} : j \in [\nu]\}$ . Choosing a function  $\phi \in \Phi$  uniformly at random yields a random (multi-)graph  $G_\phi$  with degree sequence  $\mathbf{d}$ .

It is known that conditional on  $G_\phi$  being simple, i.e. having no loops or multiple edges, it is equally likely to be any graph that has degree sequence  $\mathbf{d}$ . Also, if the maximum degree is  $O(1)$  then the probability that  $G_\phi$  is simple is bounded below by a positive quantity that is independent of  $n$ . Thus results on this model can be translated immediately to random simple graphs.

We split the REDUCE-CONSTRUCT Algorithm into the REDUCE and CONSTRUCT algorithms

which we present separately.

**Algorithm REDUCE:**

The input  $G_0 = G_\phi$  where we condition on there being no loops.

$i = \hat{\tau} = 0$ .

**While**  $G_i = (V_i, E_i) \neq (\emptyset, \emptyset)$  **do:**

**If**  $\delta(G_i) = 0$ : Perform a **vertex-0 removal**: choose a vertex of degree 0 and remove it from  $V_i$ .

**Else if**  $\delta(G_i) = 1$ : Perform a **vertex-1 removal**: choose a random vertex  $v$  of degree 1 and remove it along with its neighbor  $w$  and any edge incident to  $w$ .

**Else if**  $\delta(G_i) = 2$ : Perform a **contraction**: choose a random vertex  $v$  of degree 2. Then replace  $\{v\} \cup N(v)$  ( $v$  and its neighbors  $N(v)$ ) by a single vertex  $v_c$ . For  $u \in V \setminus (\{v\} \cup N(v))$ ,  $u$  is joined to  $v_c$  by as many edges as there are in  $G_i$  from  $u$  to  $\{v\} \cup N(v)$ . Remove any loops created.

**Else if**  $\delta(G_i) \geq 3$ : Perform a **max-edge removal**: choose a random vertex of maximum degree and remove a random edge incident with it.

**End if**

**If** the last action was a max-edge removal, say the removal of edge  $\{u, v\}$  and in the current graph we have  $d(u) = 2$  and  $u$  is joined to a single vertex  $w$  by a pair of parallel edges then perform an **auto correction contraction**: contract  $u, v$  and  $w$  into a single vertex. Remove any loops created.

**End If**

Set  $i = i + 1$  and let  $G_i$  be the current graph.

**End While**

Set  $\hat{\tau} = i$ .

Observe that we only reveal edges (pairs of the form  $(\phi(2j - 1), \phi(2j)) : j \in [\nu]$ ) of  $G_\phi$  as the need arises in the algorithm. Moreover the algorithm removes any edges that are revealed. Thus if we let  $\mathbf{d}(i)$  be the degree sequence of  $G_i$  then, given  $\mathbf{d}(i)$  and the actions performed by REDUCE until it generates  $G_i$  we have that  $G_i$  is uniformly distributed among all configurations with degree sequence  $\mathbf{d}(i)$  and no loops.

Call a contraction that is performed by REDUCE and involves only 2 vertices *bad* i.e. one where we contract  $u, v$  to a single vertex given that  $G$  contains a parallel pair of the edge  $\{u, v\}$  and  $u$  has degree 2. Otherwise call it *good*. Observe that a bad contraction can potentially be a mistake while a good contraction is never a mistake. By introducing the auto correction contraction we replace the bad contraction of the vertex set  $\{u, w\}$ , as presented in the description of REDUCE, with the good contraction of the vertex set  $\{v, u, w\}$ . Note that we do not claim that all bad contractions can be dealt with in this way. We only show later that other instance of bad contractions are very unlikely.

We now describe how we unwind the operations of REDUCE to provide us with a matching.

**Algorithm CONSTRUCT:**

**Input:**  $G_0, G_1, \dots, G_{\hat{\tau}}$  - the graph sequence produced by REDUCE, an integer  $j \in \{0, 1, \dots, \hat{\tau}\}$  and a matching  $M_j$  of  $G_j$ . (We allow the possibility of stopping REDUCE before it has finished. If we do so when  $|V(G_j)| = \Theta(n^{2/3})$  then, given that  $G_j$  has a perfect matching w.h.p., we can use the  $O(|E||V|^{1/2})$  algorithm of [13] applied to  $G_j$  to find a perfect matching  $M_j$  of  $G_j$  in  $O(n)$  time. Thereafter we can use CONSTRUCT to extend  $M_j$  to a matching of  $G_0$ .)

**For**  $i = 1$  **to**  $j$  **do:**

**If**  $\delta(G_{j-i}) = 0$ : Set  $M_{j-i} = M_{j-i+1}$

**Else if**  $\delta(G_{j-i}) = 1$ : Let  $v$  be the vertex of degree 1 chosen at the  $(j - i)$ th step of REDUCE and let  $e$  be the edge that is incident to  $v$  in  $G_{j-i}$ . Then, Set  $M_{j-i} = M_{j-i+1} \cup \{e\}$ .

**Else if**  $\delta(G_{j-i}) = 2$ : Let  $v$  be the vertex of degree 2 selected in  $G_{j-i}$ . If  $|N(v)| = 1$  i.e.  $v$  is joined to a single vertex by a double edge in  $G_{j-i}$ , set  $M_{j-i} = M_{j-i+1}$ . Else let  $N(v) = \{u, w\}$  and  $v_c$  be the new vertex resulting from the contraction of  $\{v, u, w\}$ . If  $v_c$  is not covered by  $M_{j-i+1}$  then set  $M_{j-i} = M_{j-i+1} \cup \{v, u\}$ . Otherwise assume that  $\{v_c, z\} \in M_{j-i+1}$  for some  $z \in V(G_{j-i})$ . Without loss of generality assume that in  $G_{j-i}$ ,  $z$  is connected to  $u$ . Set  $M_{j-i} = (M_{j-i+1} \cup \{\{v, w\}, \{u, z\}\}) \setminus \{v_c, z\}$ .

**Else if** :  $\delta(G_{j-i}) \geq 3$ : Set  $M_{j-i} = M_{j-i+1}$

**End For**

For a graph  $G$  and  $j \in \{0, 1, \dots, \hat{\tau}\}$  denote by  $R_0(G, j)$  and  $R_{2b}(G, j)$  the number of times that REDUCE has performed a vertex-0 removal and a bad contraction respectively until it generates  $G_j$ . For a graph  $G$  and a matching  $M$  denote by  $\kappa(G, M)$  the number of vertices that are not covered by  $M$ . The following Lemma determines the quality of the output of the REDUCE-CONSTRUCT algorithm.

**Lemma 2.** *Let  $G$  be a graph and  $M$  be the output of the Reduce-Backtrack algorithm applied to  $G$ . Then, for  $j \geq 0$ ,*

$$\kappa(G, M) = R_0(G, j) + R_{2b}(G, j) + \kappa(G_j, M_j). \quad (1)$$

*Proof.* Let  $G = G_0, G_1, \dots, G_{\hat{\tau}}$  be the sequence of graphs produced by REDUCE and let  $M_j, M_{j-1}, \dots, M_0 = M$  be the sequence of matchings produced by CONSTRUCT. Let  $R_0(G, j, i)$  and  $R_{2b}(G, j, i)$  be the number of vertex-0 removals and bad contractions performed by REDUCE going from  $G_{j-i}$  to  $G_j$ . We will prove that for every  $0 \leq i \leq j$ ,

$$\kappa(G_{j-i}, M_{j-i}) = R_0(G, j, i) + R_{2b}(G, j, i) + \kappa(G_j, M_j). \quad (2)$$

Taking  $i = j$  yields the desired result.

For  $i = 0$ , equation (1) holds as  $R_0(G, j, 0) = R_{2b}(G, j, 0) = 0$ . Assume inductively that (1) holds for  $i = k - 1$  where  $k$  satisfies  $0 < k \leq j$ . For  $i = k$ , if a max-edge deletion was performed on  $G_{j-k}$  then  $|V_{j-k}| = |V_{j-k+1}|$ . Furthermore,  $R_0(G, j, k) = R_0(G, j, k - 1)$  and  $R_{2b}(G, j, k) = R_{2b}(G, j, k - 1)$  and hence (1) continues to hold. If a vertex-0 deletion or a bad contraction was performed on  $G_{j-k}$  then  $|V_{j-k}| = |V_{j-k+1}| + 1$  and  $M_{j-k} = M_{j-k+1}$ . In the case of a vertex-0 deletion we have  $R_0(G, j, k) = R_0(G, j, k - 1) + 1$  and  $R_{2b}(G, j, k) = R_{2b}(G, j, k - 1)$  and both sides of (2) increase by one. In the case of a bad contraction we have  $R_0(G, j, k) = R_0(G, j, k - 1)$  and  $R_{2b}(G, j, k) = R_{2b}(G, j, k - 1) + 1$  and again both sides of (2) increase by one. Finally if a good contraction or a vertex-1 removal was performed on  $G_{j-k}$  then  $R_0(G, j, k) = R_0(G, j, k - 1)$  and  $R_{2b}(G, j, k) = R_{2b}(G, j, k - 1)$ . At the same time we have that  $\kappa(G_{j-i}, M_{j-i}) = \kappa(G_{j-i+1}, M_{j-i+1})$ , completing the induction.  $\square$

## 2.3 Organizing the actions taken by REDUCE

We do not analyze the effects of each action taken by REDUCE individually. Instead we group together sequences of actions, into what we call *hyperactions*, and we analyze the effects of the individual hyperactions.

We construct a sub-sequence  $\Gamma_0 = G, \Gamma_1, \dots, \Gamma_\tau$  of  $G_0, G_1, \dots, G_{\hat{\tau}}$ . Every hyperaction, starts with a max-edge removal and it consists of all the actions taken until the next max-edge removal. We let  $\Gamma_i$  be the graph that results from performing the first  $i$  hyperactions. Thus  $\Gamma_i$  is the  $i$ th graph in the sequence  $G_0, G_1, \dots, G_{\hat{\tau}}$  that has minimum degree at least 3 and going from  $\Gamma_i$  to  $\Gamma_{i+1}$  REDUCE performs a max-edge removal followed by a sequence of vertex-0, vertex-1 removals and contractions. Thus  $\Gamma_0, \Gamma_1, \dots, \Gamma_\tau$  consists of all the graphs in the sequence  $G_0, G_1, \dots, G_{\hat{\tau}}$  with minimum degree at least 3.

## 2.4 Excess and Proof Mechanics

The central quantity of this paper is the *excess* which we denote by  $ex_\ell(\cdot)$ . For a graph  $G$  and a positive integer  $\ell$  we let

$$ex_\ell(G) := \sum_{v \in V(G)} [d(v) - \ell] \mathbb{I}(d(v) > \ell).$$

### Hyperactions of Interest

For the analysis of REDUCE we consider 7 distinct hyperactions (sequences of actions) which we call hyperactions of Type 1,2,3,4,5,33 and 34 respectively. In the case that the maximum degree is larger than 3 we consider the following hyperactions: we have put some diagrams of these hyperactions at the end of the paper.

**Type 1:** A single max-edge removal,

**Type 2:** A max edge-removal followed by an auto correction contraction.

**Type 3:** A single max-edge removal followed by a good contraction.

**Type 4:** A single max-edge removal followed by 2 good contractions. In this case we add the restriction that there are exactly 6 distinct vertices  $v, u, x_1, x_2, w_1, w_2$  involved in this hyperaction and they satisfy the following: (i)  $v$  is a vertex of maximum degree, it is adjacent to  $u$  and  $\{u, v\}$  is removed during the max-edge removal, (ii)  $d(u) = d(x_1) = d(x_2) = 3$ , (iii)  $N(u) = \{v, x_1, x_2\}$ ,  $N(x_1) = \{u, x_2, w_1\}$  and  $N(x_2) = \{u, x_1, w_2\}$ . (Thus  $\{u, x_1, x_2\}$  form a triangle.) The two contractions have the same effect as contracting  $\{u, x_1, x_2, w_1, w_2\}$  into a single vertex.

In the case that the maximum degree equals 3 we also consider the following hyperactions:

**Type 5:** A max-edge removal followed by 2 good contractions that interact. In this case the 5 vertices  $u, v, x_1, x_2, z$  involved in the hyperaction satisfy the following: (i)  $\{u, v\}$  is the edge removed by the max-edge removal, (ii)  $N(v) = \{u, x_1, x_2\}$ ,  $N(u) = \{v, x_1, z\}$ , (so  $\{u, v, x_1\}$  form a triangle), (iii)  $|(N(x_1) \cup N(x_2) \cup N(z)) \setminus \{u, v, x_1, x_2, z\}| \geq 3$ . This hyperaction has the same effect as contracting all of  $\{u, v, x_1, x_2, z\}$  into a single vertex.

**Type 33:** A max-edge removal followed by 2 good contractions that do not interact. There are 6 distinct vertices involved  $v, v_1, v_2, u, u_1, u_2$ . During the max-edge removal  $\{u, v\}$  is removed. Thereafter each of the 2 sets of vertices  $\{v, v_1, v_2\}$  and  $\{u, u_1, u_2\}$  is contracted to a single vertex.

**Type 34:** A max-edge removal followed by 3 good contractions. There are 8 distinct vertices involved  $v, v_1, v_2, v, u, u_1, u_2, w_1, w_2$ . During the max-edge removal  $\{u, v\}$  is removed. The conditions satisfied by  $v, u, u_1, u_2, w_1, w_2$  and the actions that are performed on them are similar to the ones in a hyperaction of Type 4. The difference now is that  $v$  has degree 3 before the hyperaction. In addition  $\{v, v_1, v_2\}$  is contracted into a single vertex.

We divide Hyperactions of Type 3 into three classes. Assume that during a Hyperaction of Type 3 the set  $\{v, a, b\}$  is contracted,  $v$  is the contracted vertex and  $v_c$  is the new vertex. We say that such a Hyperaction is of **Type 3a** if  $d(v_c) = d(a) + d(b) - 2$ , is of **Type 3b** if  $d(v_c) = d(a) + d(b) - 4$  and is of **Type 3c** if  $d(v_c) < d(a) + d(b) - 4$ . Note that in general,  $d(v_c) = d(a) + d(b) - 2 - 2\eta_{a,b}$ , where  $\eta_{a,b}$  is the number of edges joining  $a, b$  plus the number of edges joining  $\{u, v\}$  after the max-edge removal.

With the exception of a Hyperaction of Type 3c, where  $\eta_{a,b} \geq 2$ , we refer to the Hyperactions of interest as good Hyperactions. We call any Hyperaction that is not good, including a Hyperaction of Type 3c, bad.

We next state three lemmas, whose proofs will be deferred to Section 3.

**Lemma 3.** *Let  $i \geq 0$  and assume that  $\Gamma_i$  satisfies  $ex_\ell(\Gamma_i) \leq \log^2 |V(\Gamma_i)|$  for some  $3 \leq \ell = O(1)$ . Then with probability  $1 - o(|V(\Gamma_i)|^{-1.9})$  the hyperaction that REDUCE applies to*

$\Gamma_i$  is good. In addition, it only applies a Hyperaction of Type 2, 3b or 4 with probability  $o(|V(\Gamma_i)|^{-0.9})$ .

The fact that w.h.p. we perform one hyperaction out of a small set of possibilities helps in controlling  $ex_\ell$ . Namely we have

**Lemma 4.** *Suppose that  $\Gamma_i$  satisfies  $ex_4(\Gamma) \leq \log^2 |V(\Gamma_i)|$ . Then either  $ex_4(\Gamma_i) = 0$  or*

$$\mathbb{E}(ex_4(\Gamma_{i+1}) - ex_4(\Gamma_i) | \Gamma_i) \leq -\frac{1}{3}.$$

Moreover with probability  $1 - o(|V(\Gamma_i)|^{-1.9})$  we have that

$$|ex_4(\Gamma_i) - ex_4(\Gamma_{i+1})| \leq 2.$$

**Lemma 5.** *Suppose that  $\omega \geq \log n$ . Then for every  $i \in \{0, 1, \dots, \tau\}$ ,  $|V(\Gamma_i)| \geq \omega$  implies that with probability  $1 - O(\omega^{-0.8})$ , we have  $ex_4(\Gamma_i) \leq \log^2 |V(\Gamma_i)|$ . Furthermore with probability  $1 - O(\omega^{-0.8})$  there exists  $\Gamma \in \{\Gamma_0, \Gamma_1, \dots, \Gamma_\tau\}$  that satisfies  $\omega \leq |V(\Gamma)| \leq 2\omega$  and  $ex_4(\Gamma) = 0$ .*

Lemma 3 states that as long as  $ex_\ell$  stays small then w.h.p. we perform only specific hyperactions. Lemmas 4 and 5 imply that as long we perform only those hyperactions then w.h.p.  $ex_4$  stays small. Furthermore, as none of those hyperactions consist of a vertex-0 removal or a bad contraction making we can appeal to Lemma 2. We use the above three lemmas to prove the following:

**Lemma 6.** *Let  $\Gamma_h$  be as promised by Lemma 5 that is,  $\Gamma_h \in \{\Gamma_0, \Gamma_1, \dots, \Gamma_\tau\}$  and it satisfies  $\log n \leq \omega \leq V(\Gamma_h) \leq 2\omega$  and  $ex_4(\Gamma) = 0$ . Then with probability  $1 - O(\omega^{-0.8})$ , for any matching  $M_h$  of  $\Gamma_h$ , the matching  $M$  generated by CONSTRUCT with  $j = h$ ,  $M_h$  as an input satisfies*

$$\kappa(G, M) = \kappa(\Gamma_h, M_h). \quad (3)$$

*Proof.* For  $i < h$ , Lemma 5 states that w.h.p.  $ex_4(\Gamma_i) \leq \log^2 |V(\Gamma_i)|$ . Lemma 3 then implies that for  $i < h$  with probability

$$1 - \sum_{i=1}^h o(|V(\Gamma_i)|^{-1.9}) \geq 1 - \sum_{i=1}^h o((2|E(\Gamma_i)|/3)^{-1.9}) \geq 1 - \sum_{i=\omega}^{\infty} o(2i^{-1.9}/3) = 1 - o(\omega^{-0.9})$$

only good hyperactions take place. For the first inequality we used the fact that  $\Gamma_i$  has minimum degree 3. For the second we used the fact that any Hyperaction decreases the number of edges, hence  $|E(\Gamma_i)|$  is decreasing with respect to  $i$ .

Hence w.h.p., until REDUCE generates  $\Gamma_h$  it performs no vertex-0 removals or bad contractions. Let  $\hat{h} \in \{0, 1, \dots, \hat{\tau}\}$  be such that  $G_{\hat{h}} = \Gamma_h$ . Then from the above  $R_0(G, \hat{h}) = R_{2b}(0, \hat{h}) = 0$ . Finally in this case, Lemma 2 implies that

$$\kappa(\Gamma, M_h) = \kappa(G_{\hat{h}}, M_h) = \kappa(\Gamma_h, M_h).$$

□



Observe that by a straightforward application of Lemma 6 with  $\omega = \log n$  and  $M_h$  being the matching produced by CONSTRUCT in the case  $j = \hat{\tau}$  we have that the matching  $M$  output by the REDUCE-CONSTRUCT algorithm satisfies

$$\kappa(G, M) = \kappa(\Gamma_h, h) \leq |V(\Gamma_h)| \leq 2 \log n.$$

This proves Part (a) of Theorem 1.

## 2.5 Finding a perfect matching in linear time

Lemma 6 suggests that when we run CONSTRUCT we should finish our algorithm by finding a maximum matching  $M'$  of  $\Gamma_h$ . The final ingredient for the proof of Theorem 1(b) is given by the next Theorem which implies that  $\Gamma_h$  has a perfect matching w.h.p.

**Theorem 7.** *Let  $\mathbf{d}$  be a degree sequence where  $3 \leq d(i) \leq 4$  for all  $i$ . Let  $G$  be a random graph with degree sequence  $\mathbf{d}$  and no loops. Then  $G$  has a (near)-perfect matching with probability  $1 - O(n^{-3/4})$ .*

We defer the proof of Theorem 7 to Section 4. To prove Part (b) of Theorem 1, we run algorithm *Reduce* until we find  $\Gamma_h$  as promised in Lemma 5. Here we take  $\omega = n^{2/3}$ . We know from Lemma 6 and Lemma 7 that  $\Gamma_h$  has a (near)-perfect matching with probability  $1 - O(\omega^{-0.8}) - O(\omega^{-3/4}) = 1 - O(n^{-1/2})$  that can be expanded to a (near)-perfect matching of the original graph  $G$ . Moreover since  $|V(\Gamma_h)| \leq 2\omega$ , such a matching can be found in  $O(\omega^{3/2}) = O(n)$  time. The rest of the algorithm can be executed in  $O(n)$  time and this completes the high probability part of the proof of Part (b). Also, if REDUCE-CONSTRUCT fails, then we can resort to an  $O(n^{3/2})$  time algorithm. This only happens with probability  $O(n^{-1/2})$  and this yields the claim in Part (b) about the expected running time.

## 3 Proofs of Lemmas 3, 4 and 5

**Notation 8.** *We sometimes write  $A \leq_O B$  in place of  $A = O(B)$  for aesthetic purposes.*

**Notation 9.** *Let  $K$  be an arbitrary positive integer and  $b \in \{0, 1\}$ . For a random graph  $G$  and  $v \in V(G)$ , let  $\mathcal{B}_K(G, v, b)$  be the event that  $G$  spans a subgraph that contains  $v$ , spans  $a \leq K$  vertices and  $a + b$  edges.*

We show that REDUCE either performs one of the good hyperactions given in Section 2.4 or  $\mathcal{B}_K(G, v, 1)$  occurs where  $v$  is the vertex of maximum degree chosen by REDUCE. We will also use the following standard notation: For a graph  $G$ ,  $j, \ell \in \mathbb{N}$  we let:

- $n(G) := |V(G)|$ ,  $e(G) := |E(G)$ ,
- $\delta(G)$  and  $\Delta(G)$  be the minimum and maximum degree of  $G$  respectively,

- $n_j(G)$  be the number of vertices of  $G$  of degree  $j$ ,
- $p_j(G) := \frac{jn_j(G)}{2|E(G)|}$  and  $p_{>j}(G) := \sum_{h>j} p_h(G)$ ,
- $ex_\ell(G) := \sum_{v \in V(G)} [d(v) - \ell] \mathbb{I}(d(v) > \ell)$ ,

We denote by  $n_i, e_i, n_{j,i}, p_{j,i}, p_{>j,i}$  and  $ex_{\ell,i}$  the corresponding quantities in relation to  $\Gamma_i$ .

**Lemma 10.** *Let  $K$  be an arbitrary fixed positive integer. Let  $\mathbf{d}$  be a degree sequence of length  $n$  that satisfies  $ex_\ell(G) \leq \log^2 n$  for some  $3 \leq \ell = O(1)$ . Let  $G$  be a random graph with degree sequence  $\mathbf{d}$  and no loops. Let  $b \in \{0, 1\}$ , then  $\Pr(\mathcal{B}_K(G, v, b)) = o(n^{-0.9-b})$ .*

*Proof.* Let  $G$  be a random graph with degree sequence  $\mathbf{d}$ . The fact that  $ex_\ell(G) \leq \log^2 n$  implies that  $G$  has no loops with probability bounded below by a positive constant (see for example [7]). Hence the condition of having no loops can be ignored in the proof that events have probability  $o(1)$ . Also it implies that  $\Delta = \Delta(G) \leq \ell + ex_\ell(G) \leq \ell + \log^2 n$ .

Let  $2m = \sum_{i=1}^n d(i) \leq \ell n + ex_\ell(G) = \Theta(n)$ . Then for vertex  $v$  and for  $b = 0, 1$  the probability that  $G$  spans a subgraph that covers  $v$ , spans  $a \leq K$  vertices and  $a+b$  edges can be bounded above by

$$\begin{aligned}
&\leq o \sum_{a=2}^K \binom{n}{a-1} (\Delta a)^{2(a+b)} \frac{(2m - 2(a+b))!}{2^{m-(a+b)} [m - (a+b)]!} \times \frac{2^m m!}{(2m)!} \\
&\leq o \sum_{a=2}^K n^{a-1} \Delta^{2(a+b)} \frac{m(m-1)\dots(m-(a+b)+1)}{2m(2m-1)\dots(2m-2(a+b)+1)} \\
&\leq o \sum_{a=2}^K n^{a-1} \Delta^{2(a+b)} m^{-(a+b)} \\
&= o(n^{-0.9-b}).
\end{aligned}$$

□

We will drop the subscript  $K$  from  $\mathcal{B}$ . Taking  $K = 20$  will easily suffice for the rest of the proof. Thus  $\mathcal{B}(G, v, b) = \mathcal{B}_{20}(G, v, b)$ .

### 3.1 Proof of Lemma 3

Let  $v$  be the vertex of maximum degree chosen by REDUCE and let  $u$  be the vertex adjacent to  $v$  such that  $\{u, v\}$  is chosen for removal. We will show that if  $\mathcal{B}(\Gamma_i, v, 1)$  does not occur then REDUCE performs one of the hyperactions given in Section 2.4. Also observe that if a Hyperaction of Type 2,3b,4,5 or 34 occurs then  $\mathcal{B}(\Gamma_i, v, 0)$  occurs. Lemma 10 states that  $\Pr(\mathcal{B}(\Gamma_i, v, 0)) = o(|V(\Gamma_i)|^{-0.9})$  thus proving the second part of Lemma 3. Also note that if a Hyperaction of Type 3c, occurs, corresponding to a bad Hyperaction, then  $\mathcal{B}(\Gamma_i, v, 1)$  occurs. Lemma 10 states that  $\Pr(\mathcal{B}(\Gamma_i, v, 1)) = o(|V(\Gamma_i)|^{-1.9})$ .

**Case A:**  $d(v) \geq 4$ .

If  $d(u) \geq 4$  then a hyperaction of Type 1 is performed. Thus assume  $d(u) = 3$  and consider the cases where  $|N(u)| = 1, 2, 3$ , (recall that we allow parallel edges but not self-loops).

**Case A1:**  $|N(u)| = 1$ .

$u$  is connected to  $v$  by 3 parallel edges and so  $\mathcal{B}(\Gamma_i, v, 1)$  occurs.

**Case A2:**  $|N(u)| = 2$ .

Let  $N(u) = \{v, u'\}$  and  $S = \{u, u', v\}$ . Let  $T = (N(u') \cup N(v)) \setminus S$ . If  $|T| \leq 2$  then we have  $d(S) \geq 10$  and either  $S$  spans more than 3 edges or  $S \cup T$  spans at least 7 edges. In both cases  $\mathcal{B}(\Gamma_i, v, 1)$  occurs. Assume then that  $|T| \geq 3$ . Now exactly one of  $\{u, u'\}$ ,  $\{u, v\}$  is repeated, else  $\mathcal{B}(\Gamma_i, v, 1)$  occurs. If  $\{u, u'\}$  is repeated then we perform an auto correction contraction resulting to a Hyperaction of Type 2. If  $\{u, v\}$  is repeated then we contract the remaining path  $(u', u, v)$ . Hence we have performed a Hyperaction of Type 3b.

**Case A3:**  $|N(u)| = 3$ .

Let  $N(u) = \{v, x_1, x_2\}$  and  $T = (N(x_1) \cup N(x_2)) \setminus \{u, x_1, x_2\}$ .

**Sub-case A3a:**  $|T| \leq 1$ .

$\{u, x_1, x_2\} \cup T$  spans at least  $(d(u) - 1 + d(x_1) + d(x_2) + |T|)/2 \geq 4 + |T|/2$  edges and the event  $\mathcal{B}(\Gamma_i, v, 1)$  occurs.

**Sub-case A3b:**  $T = \{w_1, w_2\}$ .

If  $v$  is at distance less than 6 from  $\{u\} \cup N(u)$  in  $\Gamma \setminus \{u, v\}$  then  $\mathcal{B}(\Gamma_i, v, 1)$  occurs. To see this consider the subgraph  $H$  spanned by  $\{u, v, x_1, x_2, w_1, w_2, y\}$  and the vertices on the shortest path  $P$  from  $v$  to  $u$  in  $\Gamma_i \setminus \{u, v\}$ . Here  $y$  is the neighbor of  $v$  on  $P$ . It must contain at least two distinct cycles. One contained in  $\{u, x_1, x_2, w_1, w_2\}$  and  $u, v, P$ . If there is no edge from  $x_1$  to  $x_2$  then  $\{v, u, x_1, x_2, w_1, w_2\}$  spans at least 7 edges and so  $\mathcal{B}(\Gamma_i, v, 1)$  occurs.

Thus we may assume that  $N(x_1) = \{u, x_2, w_1\}$ ,  $N(x_2) = \{u, x_1, w_2\}$  and  $v \notin \{w_1, w_2\} \cup N(w_1) \cup N(w_2)$ . We may also assume that  $\{w_1, w_2\}$  is not an edge of  $\Gamma$ , for otherwise  $\{u, x_1, x_2, w_1, w_2\}$  contains two distinct cycles and  $\mathcal{B}(\Gamma_i, v, 1)$  occurs. The algorithm REDUCE proceeds by contracting  $u, x_1, x_2$  into a single vertex  $x'$ .  $x'$  has degree 2 and the algorithm proceeds by performing a contraction of  $x', w_1, w_2$  into a new vertex  $w'$ . Let  $S = N\{w_1, w_2\} \setminus \{x_1, x_2\}$ . If  $|S| \leq 3$  then  $\mathcal{B}(\Gamma_i, u, 1)$  occurs. To see this observe that  $w_1, w_2$  must then have a common neighbor  $w_3$  say. Consider the subgraph  $H$  spanned by  $\{u, x_1, x_2, w_1, w_2, w_3\}$ .  $H$  contains at least 7 edges and 6 vertices. If  $|S| \geq 4$  then the new vertex has degree 4 and the sequence of actions taken by REDUCE corresponds to a hyperaction of Type 4.

**Sub-case A3c:**  $|T| \geq 3$ . After the removal of  $\{v, u\}$  we contract  $\{u, x_1, x_2\}$  into a single vertex of degree at least 3, hence a hyperaction of Type 3 is performed.

**Case B:**  $d(v) = d(u) = 3$ .

Let  $\Gamma' = \Gamma_i \setminus \{e\}$  where  $e = \{v, u\}$ .

**Case B1:** In  $\Gamma'$ ,  $u$  and  $v$  are at distance at least 4.

If  $|N(N(u))|$  and  $|N(N(v))| \leq 3$  then  $\mathcal{B}(\Gamma_i, v, 1)$  occurs. Thus we can assume that either  $|N(N(u))| = 4$  and/or  $|N(N(v))| = 4$ . If both  $|N(N(u))|, |N(N(v))| = 4$  then *Reduce* will

perform 2 good contractions and this amounts to a hyperaction of Type 33. Assume then that  $|N(N(u))| = 4$  and that  $|N(N(v))| \leq 3$ . If  $|N(N(v))| = 3$  then again *Reduce* will perform 2 good contractions amounting to a hyperaction of Type 33. If  $|N(N(v))| = 2$  and so  $v$  is in a triangle then *Reduce* will perform a hyperaction of Type 34. Finally, if  $|N(N(v))| = 1$  then  $\mathcal{B}(\Gamma_i, v, 1)$  occurs.

**Case B2: In  $\Gamma'$ ,  $u$  and  $v$  are at distance 3.**

In  $\Gamma_i$  there is a cycle  $C$  of length 4 containing  $u, v$ . If in  $\Gamma'$  we find that  $|N(N(u))| \leq 3$  or  $|N(N(v))| \leq 3$  or  $|N(u) \cap N(N(v))| > 1$  or  $|N(v) \cap N(N(u))| > 1$  then  $\mathcal{B}(\Gamma_i, v, 1)$  occurs. This is because the graph spanned by  $\{u, v\} \cup N(u) \cup N(v) \cup N(N(u) \cup N(N(v)))$  in  $\Gamma_i$  will contain a cycle distinct from  $C$ . Assume this is not the case. Then after the max-edge removal of  $\{u, v\}$  we have a contraction of  $\{u\} \cup N(u)$  followed by a contraction of  $\{u\} \cup N(u)$ . Observe that neither contraction reduces the size of  $N(N(u))$  or  $N(N(v))$ . Thus REDUCE performs a hyperaction of Type 33.

**Case B3: In  $\Gamma'$ ,  $u$  and  $v$  are at distance 2.**

In the case that  $u, v$  have 2 common neighbors in  $\Gamma'$  we see that  $\mathcal{B}(\Gamma_i, v, 1)$  occurs. Assume then that they have a single common neighbor  $x_1$ . Let  $z, x_2$  be the other neighbors of  $u, v$  respectively. Then either  $\mathcal{B}(\Gamma_i, v, 1)$  occurs or REDUCE performs a hyperaction of Type 5.

**Case B4: In  $\Gamma'$ ,  $u$  and  $v$  are at distance 1.**

So here we have that  $\{u, v\}$  is a double edge in  $\Gamma_i$ . Let  $x, y$  be the other neighbors of  $u, v$  respectively in  $\Gamma$ . Assuming that  $\mathcal{B}(\Gamma_i, v, 1)$  does not occur, REDUCE performs a max-edge removal followed by a single good contraction and this will be equivalent to a hyperaction of Type 3, involving the contraction of one of  $x, u, v$  or  $u, v, y$ .  $\square$

## 3.2 Proof of Lemma 4

The inequality  $ex_{4,i} \leq \log^2 n_i$  implies that  $p_{j,i} \leq \log^2 n_i / 2e_i = o(n_i^{-0.95})$  for  $5 \leq j \leq \Delta$ . It also implies that the maximum degree  $\Delta$  of  $\Gamma$  satisfies  $\Delta = O(\log^2 n)$ .

In the case that  $\mathcal{B}(G, v, 1)$  occurs, i.e. with probability  $O(n_i^{-1.9})$  (see Lemma 3)

$$|ex_{4,i} - ex_{4,i+1}| \leq 2e_i \leq 4n_i + ex_{4,i} \leq 5n_i.$$

Observe that if an action of Type 2,3b,4,5 or 34 takes place then  $v$  lies on a subgraph with  $a < 12$  vertices and  $a$  edges. Lemma 3 states that this occurs with probability  $o(n^{-0.9})$ . For all the above hyperactions we have  $|ex_{4,i+1} - ex_{4,i}| \leq 2$ . This follows from the fact that performing a contraction can increase  $ex_4$  by at most 2. This is because the initial vertices with degrees say  $2, d_1, d_2$  contributed  $\max\{0, (d_1 - 4) + (d_2 - 4)\}$  to  $ex_{4,i}$  while the new contracted vertex has degree  $d_1 + d_2 - 2$  and contributes  $\max\{0, d_1 + d_2 - 2 - 4\}$  to  $ex_{4,i+1}$ . Moreover observe that if a hyperaction of Type 5, 33 or 34 occurs then all the vertices involved have degree 3. If  $\mathcal{B}(\Gamma, v, 1)$  does not occur then a hyperaction of Type 5 will increase  $ex_{4,i}$  by 1 (since there will be one new vertex of degree 5). Hyperactions of Type 33 or Type 34 do not change  $ex_4$ . Thus it remains to examine the effects of a hyperaction of Type 1 or of Type 3a.

If  $ex_{4,i} = 0$  then a hyperaction of Type 1 does not increase  $ex_{4,i}$  while a hyperaction of Type 3 could increase it by 2.

If  $ex_4(\Gamma) > 0$  then the  $i^{\text{th}}$  hyperaction starts with a max-edge removal.

**Case 1:** If the smaller degree vertex involved is of degree larger than 3, then this results in a hyperaction of Type 1. This happens with probability  $(1 + o(1))(1 - p_{3,i})$ . Furthermore in this case  $ex_{4,i} - 2 \leq ex_{4,i+1} \leq ex_{4,i} - 1$ . (The  $(1 + o(1))$  factor arises because of  $O(1)$  degree changes during the hyperaction.)

**Case 2:** If the smaller degree vertex  $v$  involved is of degree 3 then a contraction is performed. And this occurs with probability  $(1 + o(1))p_{3,i}$ . If the contraction results in a vertex of degree at least 3 then we have a hyperaction of Type 3a, and not of Type 1. Also this is the only way for a hyperaction of Type 3a to occur. Let the other two vertices involved in the contraction be  $a, b$  and have degrees  $d_a, d_b$  respectively. Now  $d_a = d_b = 3$  with probability  $(1 + o(1))p_{3,i}^2$ , resulting in a new vertex that has degree at most 4. In this case,  $ex_{4,i+1} - ex_{4,i} = -1$  (the -1 here originates from the max-edge removal). Else if  $d_a = 3, d_b = 4$  then we have  $-1 \leq ex_{4,i} - ex_{4,i} \leq -1 + 1 = 0$ . And this occurs with probability  $(1 + o(1))2p_{3,i}p_{4,i}$ . Else if  $d_a = d_b = 4$  then we have  $-1 \leq ex_{4,i+1} - ex_{4,i} \leq -1 + 2 = 1$  (this occurs with probability  $p_{3,i}p_{4,i}^2$ ). Otherwise a vertex of degree at least 5 is involved and given our upper bound on  $ex_{4,i}$ , this happens with probability  $o(1)$ . If the event  $\mathcal{B}(\Gamma, v, 1)$  does not occur then the new contracted vertex has degree  $d_a + d_b - 2$ . Hence  $-1 \leq ex_{4,i} - ex_{4,i} \leq -1 + 2 = 1$ .

Thus in all cases  $|ex_{4,i} - ex_{4,i}| \leq 2$  and if  $ex_{4,i} > 0$  then

$$\begin{aligned} \mathbb{E}[ex_{4,i+1} - ex_{4,i} | \Gamma_i] &\leq -(5n_i) \cdot o(n_i^{-1.9}) + 2n_i^{-0.95} - (1 - p_{3,i}) - p_{3,i}^3 + p_{3,i}p_{4,i}^2 + o(1) \\ &\leq -(1 - p_{3,i}) - p_{3,i}^3 + p_{3,i}(1 - p_{3,i})^2 + o(1) \leq -\frac{1}{3} \end{aligned}$$

(The final expression in  $p_{3,i}$  is maximized at  $p_{3,i} = 1/2$ , for  $p_{3,i} \in [0, 1]$ .) □

### 3.3 Proof of Lemma 5

We start by proving the following Lemma:

**Lemma 11.** *Let  $\Gamma_h$  be such that  $ex_{4,h} = 0$ . Then with probability  $1 - o(n_j^{-1.8})$  there exists  $1 \leq j \leq 0.25 \log^2 n_h$  satisfying  $ex_{4,h+j} = 0$ . Furthermore  $ex_{4,h+i} \leq \log^2 n_{h+i}$  for  $i \leq j$ .*

*Proof.* If  $ex_{4,h+1} = 0$  then we are done. Otherwise Lemma 4 implies that  $ex_{4,h+1} \in \{1, 2\}$  with probability  $1 - o(n_h^{-1.9})$ .

Let  $\mathcal{E}_T$  be the event that for  $j \leq 0.25 \log^2 n_h$  REDUCE performs only hyperactions of Type 1,2,3,4,5,33 or 34. Such a hyperaction reduces the vertex set by at most 8. Lemma 3 implies that  $\mathcal{E}_T$  occurs with probability  $1 - o(n_h^{-1.8})$ . Moreover if  $\mathcal{E}_T$  occurs for  $j < 0.25 \log^2 n_h$  then  $|n_{h+j}| \geq n_h - 8 \cdot 0.25 \log^2 n_h$ . In addition from Lemma 4 we have that with probability  $1 - o(n^{-1.8})$ ,  $|ex_{4,h+j} - ex_{4,h+j-1}| < 2$  for  $j < 0.25 \log^2 n$  hence  $ex_{4,h+j} \leq 2 \cdot 0.25 \log^2 n_h \leq \log^2 n_{h+j}$ . Finally conditioned on  $\mathcal{E}_T$  since  $ex_{4,h+1} = 1$  or  $2$ , the probability that there is no  $2 \leq j \leq 0.25 \log^2 n_h$  satisfying  $ex_{4,h+j} = 0$  is bounded by the probability that the sum of

$0.25 \log^2 n - 2$  independent random variables with magnitude at most 2 and expected value smaller than  $-1/3$  (see Lemma 4) is positive. From Hoeffding's inequality [11] we have that the later probability is bounded by  $\exp \left\{ -\frac{2(\frac{1}{3} \log^2 n_h - 3)^2}{\log^2 n_h} \right\} = o(n_h^{-2})$ .  $\square$

Now let  $\Gamma_0, \Gamma_{i_1}, \dots, \Gamma_{i_\ell}$  be the subsequence of  $\Gamma_0, \Gamma_1, \dots, \Gamma_\tau$  that includes all the graphs that have  $ex_4 = 0$  and at least  $\omega$  vertices. Then since  $\Gamma_i$  has minimum degree 3 and  $e_i$  is decreasing with respect to  $i$  using Lemma 11 we have that with probability

$$1 - \sum_{i:n_i \geq \omega} o(n_i^{-1.8}) \geq 1 - \sum_{i:n_i \geq \omega} 2e_i^{-1.8}/3 \geq 1 - \sum_{i=\omega}^{\infty} i^{-1.8} = 1 - o(\omega^{-0.8})$$

we have that for  $j < \ell$  we have  $n_{i_j} - n_{i_{j+1}} \leq 8 \cdot 0.25 \log^2 n_{i_j} = 2 \log^2 n_{i_j}$  and all the graphs  $\Gamma_i$  preceding  $\Gamma_{i_\ell}$  in  $\Gamma_0, \Gamma_1, \dots, \Gamma_\tau$  satisfy  $ex_{4,i} \leq \log^2 n_i$ . Suppose now that  $n_{i_\ell} > 2\omega$ . Then the above argument implies that w.h.p. there is  $j > i_\ell$  such that  $ex_{4,j} = 0$  and  $n_j \geq n_{i_\ell} - 2 \log^2 n_{i_\ell} \geq \omega$  and this contradicts the definition of  $i_\ell$ . Thus, w.h.p.,  $\omega \leq n_{i_\ell} \leq 2\omega$  and this completes the proof of Lemma 5.  $\square$

## 4 Existence of a Perfect Matching

We devote this section to the proof of Lemma 7. As discussed in the previous section it is enough to prove that given a degree sequence  $\mathbf{d} = (d(1), \dots, d(n))$  consisting only of 3's and 4's, if we let  $G$  be a random configuration (multi)-graph with degree sequence  $\mathbf{d}$ , then w.h.p.  $G$  has a (near)-perfect matching (i.e we can lift the condition that  $G$  has no loops). We will first assume that  $n$  is even and verify Tutte's condition. That is for every  $W \subset V$  the number of odd components induced by  $V \setminus W$ ,  $q(V/W)$ , is not larger than  $|W|$ . We split the verification of Tutte's condition into two lemmas.

**Lemma 12.** *Let  $W \subset V$  be a set of minimum size that satisfies  $q(V/W) > |W|$ . Then with probability  $1 - O(n^{-3/4})$ ,  $|W| > 10^{-5}n$ .*

**Lemma 13.** *Let  $W \subset V$  be a set of maximum size that satisfies  $q(V/W) > |W|$ . Then with probability  $1 - O(n^{-3/4})$ ,  $|W| < 10^{-5}n$ .*

Lemmas 12 and 13 together imply Tutte's condition as there exists no set with size that is both strictly larger and strictly smaller than  $10^{-5}n$ . In the proof of these lemmas we use the following estimates.

**Lemma 14.** *The number of distinct partitions of a set of size  $2r$  into 2-element subsets, denoted by  $\phi(2r)$ , satisfies  $\phi(2r) = \Theta \left( \left( \frac{2r}{e} \right)^r \right)$ . Also for  $\ell < r$  we have  $\phi(2r) \leq 2r^{r-\ell} \left( \frac{2\ell}{e} \right)^\ell$ .*

*Proof.* To generate a matching we first choose a permutation of the  $2r$  items and then we pair the  $(2i - 1)$ th item with the  $2i$ th item. Therefore, using Stirling's approximation we have

$$\phi(2r) = \frac{(2r)!}{2^r r!} = \frac{\Theta(\sqrt{2r} \left( \frac{2r}{e} \right)^{2r})}{\Theta(2^r \sqrt{r} \left( \frac{r}{e} \right)^r)} = \Theta \left( \left( \frac{2r}{e} \right)^r \right).$$

Also

$$\phi(2r) = \frac{(2r)!}{2^r r!} \leq \frac{(2r)^{r-\ell} (2\ell)!}{2^{r-\ell} 2^\ell \ell!} \leq 2r^{r-\ell} \left(\frac{2\ell}{e}\right)^\ell,$$

where we have used  $(2r)! \leq (2r)^{2(r-\ell)} (2\ell)!$ . □

#### 4.1 Proof of Lemma 12:

Let  $W$  be a set satisfying  $q(V \setminus W) > |W|$  of minimum size and assume  $2 \leq w = |W| \leq 10^{-5}n$ . We can rule out the case  $w = 1$  from the fact that with probability  $1 - O(1/n)$ ,  $G$  will be 3-connected, see e.g. the proof of Theorem 10.8 in [7]. Let  $C_z$  be a component spanned by  $V \setminus W$  of maximum size and let  $r = |C_z|$ .

**Case 1:**  $r = |C_z| \leq 0.997n$ . In this case we can partition  $V \setminus W$  into two parts  $V_1, V_2$  such that (i) each  $V_l, l = 1, 2$  is the union of components of  $V \setminus W$ , (ii)  $|V_1| \geq |V_2|$ , and (iii)  $|V_2| \geq (n - (r + w))/2 \geq 10^{-3}n$ .

Let  $d_2 = d(V_2)$  and  $d_W = d(W)$ . Out of the  $d_W$  endpoints in  $W$  (i.e. configuration points that correspond to vertices in  $W$ ),  $\ell \leq d_W$  are matched with endpoints in  $V_2$  and the rest with endpoints in  $V_1$ .

For fixed  $i, w, d_W$  the probability that there are sets  $V_1, V_2, W$  with  $w = |W|, d(W) = d_W$  and  $|V_2| = i$  satisfying  $1 \leq w \leq 10^{-5}n, 10^{-3}n \leq i \leq 0.5n$  and  $d_W \leq 4w \leq 0.04i$ , such that  $V_1 \times V_2$  spans no edges is bounded by

$$\begin{aligned} p_1 &\leq \sum_{\ell=0}^{d_W} \binom{n}{i} \binom{n-i}{w} \binom{d_W}{\ell} \frac{\phi(d_2 + \ell) \cdot \phi(2m - d_2 - \ell)}{\phi(2m)} \\ &\leq o \sum_{\ell=0}^{d_W} \left(\frac{en}{i}\right)^i \left(\frac{en}{w}\right)^w 2^{d_W} \frac{\left(\frac{d_2 + \ell}{e}\right)^{(d_2 + \ell)/2} \left(\frac{2m - d_2 - \ell}{e}\right)^{(2m - d_2 - \ell)/2}}{\left(\frac{2m}{e}\right)^m} \\ &\leq \sum_{\ell=0}^{d_W} \left(\frac{en}{i}\right)^i \left(\frac{100en}{i}\right)^{i/100} 2^{i/25} \left(\frac{d_2 + \ell}{2m}\right)^{(d_2 + \ell)/2} \left(1 - \frac{d_2 + \ell}{2m}\right)^{(2m - d_2 - \ell)/2} \\ &\leq o \sum_{\ell=0}^{d_W} \left(\frac{1600(en)^{101}}{i^{101}}\right)^{i/100} \left(\frac{d_2 + \ell}{2m}\right)^{(d_2 + \ell)/2} \exp\left\{-\frac{d_2 + \ell}{2} \left(1 - \frac{d_2 + \ell}{2m}\right)\right\} \end{aligned}$$

For the third line we used the fact that  $w \leq i/100$  and  $d_W \leq 4w \leq i/25$ .

Let  $f(x) = x^x e^{-x(1-x)}$  and  $L(x) = \log f(x)$ . Then  $L''(x) = x^{-1} + 2$  and so  $L$  and hence  $f$  is convex for  $x > 0$ . Now  $d_2 + \ell \in J = [3i, 4.04i]$  and since  $\left(\frac{d_2 + \ell}{2m}\right)^{(d_2 + \ell)/2} \exp\left\{-\frac{d_2 + \ell}{2} \left(1 - \frac{d_2 + \ell}{2m}\right)\right\} = f\left(\frac{d_2 + \ell}{2m}\right)^m$  we see that its maxima are at the endpoints of  $J$ . In general  $3i \leq 3n/2 \leq m$ . However when  $d_2 + \ell = 4.04i$  we have that

$$2m \geq 4.04i + 3(n - i - w) \geq 4.04i + 3(n - 1.01i) = 3n + 1.01i. \quad (4)$$

**Case 1a:**  $d_2 + \ell = 3i$ .

We have  $\frac{d_2 + \ell}{2m} \leq \frac{3i}{3n} \leq \frac{1}{2}$  and  $(d_2 + \ell)(1 - \frac{d_2 + \ell}{2m}) \geq 3i/2$ . Therefore,

$$\begin{aligned} p_1 &\leq_O w \left( \frac{1600(en)^{101}}{i^{101}} \right)^{i/100} \left( \frac{i}{n} \right)^{3i/2} e^{-3i/4} \\ &= w \left[ \frac{1600e^{26}}{2^{49}} \left( \frac{2i}{n} \right)^{49} \right]^{i/100} \\ &\leq w \left[ e^{-1/2} \left( \frac{2i}{n} \right)^{49} \right]^{i/100} \\ &\leq w e^{-i/200}. \end{aligned}$$

**Case 1b:**  $d_2 + \ell = 4.04i$ .

It follows from (4) that  $\frac{d_2 + \ell}{2m} \leq \frac{4.04i}{3n + 1.01i} \leq 0.577$  where the second inequality uses  $i \leq n/2$ . It follows from this that  $(d_2 + \ell)(1 - \frac{d_2 + \ell}{2m})/2 \geq 0.85i$ . Hence,

$$\begin{aligned} p_1 &\leq_O w \left( \frac{1600(en)^{101}}{i^{101}} \right)^{i/100} \left( \frac{4.04i}{3n + 1.01i} \right)^{2.02i} e^{-0.85i} \\ &\leq_O w \left[ 1600e^{16} \left( \frac{n}{i} \right)^{101} \left( \frac{4.04i}{3n} \right)^{101} \left( \frac{4.04i}{3n + 1.01i} \right)^{101} \right]^{i/100} \\ &\leq_O w \left[ 1600e^{16} \left( \frac{4.04}{3} \cdot 0.577 \right)^{101} \right]^{i/100} \\ &\leq_O w e^{-i/100}. \end{aligned}$$

Therefore the probability that Case 1 is satisfied is bounded by a constant times

$$\sum_{w=1}^{10^{-5}n} \sum_{i=10^{-3}n}^{0.5n} w e^{-i/200} = O(n^{-3/4}).$$

**Case 2:**  $r = |C_z| \geq 0.997n$ . Let  $V_1 = V(C_z)$ ,  $V_2 = V \setminus (V_1 \cup W)$ . First note that  $V_2$  spans at least  $w$  components. Therefore  $|V_2| \geq w$ . To lower bound  $e(V_2 : W)$  we use the following Claim.

**Claim 1** Every vertex in  $W$  is adjacent to at least 3 distinct components in  $V \setminus W$ , and hence to at least 2 vertices in  $V_2$ .

**Proof of Claim 1:** Let  $v \in W$  be such that it is adjacent to  $t \in \{0, 1, 2\}$  components in  $V \setminus W$ . Consider  $W' = W \setminus \{v\}$ . Thus  $|W'| = |W| - 1$ . If  $t = 0$  then  $q(V \setminus W') = q(V \setminus W) + 1$ . If  $t = 1$  then  $q(V \setminus W') \geq q(V \setminus W) - 1$ . If  $t = 2$  then if the both of the corresponding components have odd size then the new component will also have odd size, while if only one of them has odd size then the new one has even size. Finally if both have even size the new one has odd size. In all three cases the inequality  $q(V \setminus W') \geq q(V \setminus W) - 1$  is satisfied. Therefore  $q(V \setminus W') \geq q(V \setminus W) - 1 > |W| - 1 = |W'|$  contradicting the minimality of



$W$ . □

From Claim 1 it follows that  $W : V_2$  spans at least  $2w$  edges. We also have that  $|V_2| \leq n - r - w \leq 0.003n$ . For fixed  $2 \leq w \leq 10^{-5}n$ ,  $3w \leq d_W \leq 4w$  and  $w \leq i$  the probability that there exist such sets  $V_1, V_2, W$ ,  $|V_2| = i$ ,  $w = |W|$ ,  $d(W) = d_W$  and  $2w \leq \ell = e(V_2 : W) \leq 4w$  is bounded by

$$\begin{aligned}
& \sum_{\ell=2w}^{d_W} \binom{n}{i} \binom{n-i}{w} \binom{d_W}{\ell} \frac{\phi(d_2 + \ell) \cdot \phi(2m - d_2 - \ell)}{\phi(2m)} \\
& \leq_o \sum_{\ell=2w}^{d_W} \left(\frac{en}{i}\right)^i \left(\frac{en}{w}\right)^w 2^{4w} \frac{\left(\frac{d_2+\ell}{2}\right)^{(d_2+\ell-2w)/2} \left(\frac{2w}{e}\right)^w \left(\frac{2m-d_2-\ell}{e}\right)^{(2m-d_2-\ell)/2}}{\left(\frac{2m}{e}\right)^m} \\
& = \sum_{\ell=2w}^{d_W} \left(\frac{en}{i}\right)^i \left(\frac{en}{w}\right)^w 2^{4w} \left(\frac{2w}{2m}\right)^w \left(\frac{d_2+\ell}{2m} \cdot \frac{e}{2}\right)^{(d_2+\ell-2w)/2} \left(1 - \frac{d_2+\ell}{2m}\right)^{(2m-d_2-\ell)/2} \\
& \leq_o w \left(\frac{en}{i}\right)^i \left(\frac{16e}{3}\right)^w \left(\frac{5i}{3n} \cdot \frac{e}{2}\right)^{3i/2} \\
& \leq_o w \left(e^2 \left(\frac{16e}{3}\right)^{2w/i} \frac{5^3 i}{3^3 n}\right)^{i/2}.
\end{aligned}$$

For the second line we used the second inequality of Lemma 14. For the fourth line we used that  $2w \leq \ell$ ,  $d_2 + \ell \leq 4|V_2| + 4w \leq 0.01204n$  and so  $\left(\frac{d_2+\ell}{2m} \cdot \frac{e}{2}\right)^{(d_2+\ell-2w)/2}$  is maximized when  $d_2, \ell$  are as small as possible, that is  $d_2 = 3i, \ell = 2w$ . Furthermore note that  $d_2 + \ell - 2w \geq d_2 \geq 3i$  and  $i \geq q(V \setminus W) - 1 \geq w$ . Therefore the probability that Case 2 is satisfied is bounded by a constant times

$$\begin{aligned}
& \sum_{w=2}^{10^{-5}n} \sum_{i=w}^{0.003n} \left(e^2 \left(\frac{16e}{3}\right)^{2w/i} \frac{5^3 i}{3^3 n}\right)^{i/2} \\
& \leq_o \sum_{w=2}^{10^{-5}n} \sum_{i=w}^{2w} \left(\frac{C_1 i}{n}\right)^{i/2} + \sum_{w=2}^{10^{-5}n} \sum_{i=2w}^{0.003n} \left(\frac{C_2 i}{n}\right)^{i/2}
\end{aligned}$$

where  $C_1 = 16^2 5^3 e^4 / 3^5$ ,  $C_2 = 16 \cdot 5^3 e^3 / 3^4$ ,

$$\begin{aligned}
& \leq \sum_{i=2}^{n^{1/4}} i \left( \left(\frac{C_1}{n^{3/4}}\right)^{i/2} + \left(\frac{C_2}{n^{3/4}}\right)^{i/2} \right) + \sum_{i=n^{1/4}}^{2 \cdot 10^{-5}n} i \left(\frac{2C_1}{10^5}\right)^{i/2} + \sum_{i=n^{1/4}}^{0.003n} i \left(\frac{6C_2}{10^3}\right)^{i/2} \\
& = O(n^{-3/4}).
\end{aligned}$$

Finally, since  $G$  has an even number of vertices, for  $W = \emptyset$  we have  $|W| = q(V \setminus W) = 0$ . □

## 4.2 Proof of Lemma 13:

Let  $W$  be a set satisfying  $q(V \setminus W) > |W|$  of maximum size and assume  $w = |W| \geq 10^{-5}n$ .

**Claim 2** No component induced by  $V \setminus W$  is a tree with more than one vertex.

**Proof of Claim 2:** Indeed assume that  $C_i$  is such a component. If  $|C_i|$  is even then let  $v$  be a leaf of  $C_i$  and define  $W' = W \cup \{v\}$ . Then  $C_i \setminus \{v\}$  is an odd component in  $V \setminus W'$  and  $q(V \setminus W') = q(V \setminus W) + 1 > |W| + 1 = |W'|$  contradicting the maximality of  $W$ .

Thus assume that  $|C_i|$  is odd. Let  $L_1$  be the set of leaves of  $C_i$  and  $L_2$  be the neighbors of  $L_1$ . Set  $W' = W \cup |L_1|$ . Then  $|L_1| \geq |L_2|$ . Furthermore every vertex in  $L_1$  is an odd component in  $V \setminus W'$  and in the case  $|L_1| = |L_2|$  then  $C_i \setminus (L_1 \cup L_2)$  is also an odd component in  $V \setminus W'$ . Therefore,

$$\begin{aligned} q(V/W') &= q(V/W) - 1 + |L_1| + \mathbb{I}(|L_1| = |L_2|) \\ &\geq q(V/W) + |L_2| + |L_1| - |L_2| + \mathbb{I}(|L_1| = |L_2|) - 1 \\ &> |W| + |L_2| = |W'|, \end{aligned}$$

contradicting the maximality of  $W$ . □

We partition  $V \setminus W$  into three sets,  $W_1, W_2$  and  $W_3$ , as follows. With the underlying graph being the one spanned by  $V \setminus W$ ,  $W_1$  consists of the isolated vertices in  $V \setminus W$ ,  $W_2$  consists of the vertices spanned by components that contain a cycle and have size  $s \leq \frac{1}{10} \log n$  and  $W_3$  consists of the vertices that are spanned by a component of size at least  $\frac{1}{10} \log n$ . Finally let  $W_4 = W_2 \cup W_3$ . To lower bound  $W_1$  we use the following claim.

**Claim 3:** W.h.p.  $W_4$  spans at most  $\frac{11w}{\log n}$  components in  $V \setminus W$ .

**Proof of Claim 3:** First observe that the number of components spanned by  $W_2$  is smaller than the number of cycles of size at most  $\frac{1}{10} \log n$  in  $G$ , which we denote by  $r$ .

$$\begin{aligned} \Pr(r \geq n^{0.3}) &\leq n^{-0.3} \sum_{q=1}^{0.1 \log n} \binom{n}{q} 4^q q! \frac{\phi(2q)\phi(2m-2q)}{\phi(2m)} \\ &\leq_o n^{-0.3} \sum_{q=1}^{0.1 \log n} \left(\frac{en}{q}\right)^q 4^q \left(\frac{2q}{e}\right)^q \left(\frac{e}{2m}\right)^q \\ &\leq_o n^{-0.3} \sum_{q=1}^{0.1 \log n} \left(\frac{8e}{3}\right)^q \leq_o n^{-0.3} (\log n) 8^{0.1 \log n} = o(1). \end{aligned}$$

Hence w.h.p.  $W_2$  spans at most  $n^{0.3}$  components. Moreover every component spanned by  $W_3$  has size at least  $\frac{1}{10} \log n$ . Therefore  $W_4$  spans at most  $n^{0.3} + \frac{10w}{\log n} = \frac{(1+o(1))10w}{\log n}$  components in  $V \setminus W$ . □

Since  $W_4$  spans at most  $u = \frac{11w}{\log n}$  components in  $V \setminus W$  and no component is a tree it follows that the rest of the components consist of at least  $q(V \setminus W) - u > w - u$  isolated vertices that lie in  $W_1$ .

For convenience, we move  $|W_1| - (w - u)$  vertices from  $W_1$  to  $W_4$ . Therefore  $|W_1| = w - u$ . Let  $k_1$  be the number of vertices of degree 4 in  $W_1$  and  $d = d(W) - d(W_1)$ . Then  $0 \leq d \leq$

$4w - (3(w - u) + k_1) = w + 3u - k_1$ . For fixed  $10^{-5}n \leq w \leq 0.5n$  the probability that there exist such sets  $W, W_1, W_4$  is bounded by

$$p_2 \leq \sum_{k_1=0}^{w-u} \sum_{d=0}^{w+3u-k_1} \binom{n}{2w} \binom{2w}{w} \binom{w}{u} \binom{4w}{d} \Pr(d(W) - d(W_1) = d) \quad (5)$$

$$\times (3(w - u) + k_1)! \times \frac{[2m - [6(w - u) + 2k_1]]!}{2^{m - [3(w - u) + k_1]} [m - [3(w - u) + k_1]]!} \times \frac{2^m m!}{(2m)!}. \quad (6)$$

**Explanation:** We first choose the sets  $W, W_1$  and  $W_4$  of size  $w, w - u$  and  $n - 2w + u$  respectively. This can be done in  $\binom{n}{2w} \binom{2w}{w} \binom{w}{u} \binom{n-2w+u}{u}^{-1}$  ways, but we ignore the final factor.

From the at most  $4w$  copies of vertices in  $W$  we choose a set  $W'' \subset W$  be of size  $d$ . We let  $W' = W/W''$ . These are the copies of vertices that will be matched with those in  $W_1$ .

In the calculations that follow we let  $a = w/n \geq 10^{-5}$ . We also let  $k_4$  be the number of vertices of degree 4 that lie in  $W_4$ . We first bound the binomial coefficients, found in the first line.

$$\begin{aligned} \binom{n}{2w} \binom{2w}{w} \binom{w}{u} \binom{4w}{d} &= \binom{n}{2an} \binom{2an}{an} \binom{an}{u} \binom{4an}{d} \\ &\leq 2^{o(n)} \left(\frac{1}{2a}\right)^{2an} \left(\frac{1}{1-2a}\right)^{(1-2a)n} 2^{2an} \left(\frac{4ean}{d}\right)^d \\ &= 2^{o(n)} \left(\frac{1}{a}\right)^{2an} \left(\frac{1}{1-2a}\right)^{(1-2a)n} \left(\frac{4ean}{d}\right)^d. \end{aligned} \quad (7)$$

For the second line we used that  $u \leq u_0$  which implies that  $\binom{an}{u} = 2^{o(n)}$ . Observe that

$$2m = 6(w - u) + 2k_1 + d + 3(n - 2w + u) + k_4 = 3n + d + 2k_1 + k_4 - 3u. \quad (8)$$

Let  $m_0 = d + 2k_1 + k_4 - 3u$ . For the terms in line (6) we have

$$\begin{aligned} \frac{(2m)!}{2^m m!} &= \frac{(3n)!}{2^{1.5n} (1.5n)!} \frac{\prod_{i=1}^{m_0} (3n + i)}{2^{m_0/2} \prod_{i=1}^{m_0/2} (1.5n + i)} \geq o \left(\frac{3n}{e}\right)^{1.5n} \prod_{i=1}^{m_0/2} [3n + (2i - 1)]. \\ &\geq \left(\frac{3n}{e}\right)^{1.5n} e^{-o(n)} (3n)^{-3u/2} \prod_{i=1}^{d/2+k_1+k_4/2} [3n + (2i - 1)] \end{aligned}$$

Equation (8) implies that

$$2m - [6(w - u) + 2k_1] = 3(1 - 2a)n + 3u + k_4 + d.$$

Thus,

$$\begin{aligned}
& \frac{[2m - [6(w - u) + 2k_1]]!}{2^{m - [3(w - u) + 2k_1]} [m - [3(w - u) + k_1]]!} = \frac{[3(1 - 2a)n]!}{2^{1.5(1-2a)n} [1.5(1 - 2a)n]!} \cdot \frac{\prod_{i=1}^d 3(1 - 2a)n + i}{2^{d/2} \prod_{j=1}^{d/2} 1.5(1 - 2a)n + j} \\
& \times \frac{\prod_{i=1}^{k_4} 3(1 - 2a)n + d + i}{2^{k_4/2} \prod_{j=1}^{k_4/2} 1.5(1 - 2a)n + d/2 + j} \cdot \frac{\prod_{i=1}^{3u} 3(1 - 2a)n + d + k_4 + i}{2^{3u/2} \prod_{j=1}^{3u/2} 1.5(1 - 2a)n + d/2 + k_4/2 + j} \\
& \leq_o \left( \frac{3(1 - 2a)n}{e} \right)^{1.5(1-2a)n} \prod_{i=1}^{d/2} [3(1 - 2a)n + (2i - 1)] \prod_{j=1}^{k_4/2} [3(1 - 2a)n + d + (2j - 1)] \cdot (2m)^{3u/2} \\
& \leq_o \left( \frac{3(1 - 2a)n}{e} \right)^{1.5(1-2a)n} [3(1 - 2a)n + an/2]^{d/2} (2m)^{3u/2} \prod_{j=1}^{k_4/2} [3(1 - 2a)n + d + (2j - 1)]
\end{aligned}$$

For the last inequality we used the Arithmetic Mean-Geometric Mean inequality and the fact that  $d/2 \leq an/2 + o(n)$ , which follows from  $d \leq w + 3u - k_1$ .

For the first term of (6) we have

$$[3(w - u) + k_1]! \leq \frac{3w!}{(3(w - u))^{3u}} \prod_{i=1}^{k_1} (3(w - u) + i) \leq \left( \frac{3an}{e} \right)^{3an} \frac{2^{o(n)}}{n^{3u}} \prod_{i=1}^{k_1} (3(w - u) + i).$$

Thus the expression in (6) is bounded by

$$\begin{aligned}
& 2^{o(n)} \left( \frac{3an}{e} \right)^{3an} \frac{1}{n^{3u}} \prod_{i=1}^{k_1} (3(w - u) + i) \\
& \times \left( \frac{3(1 - 2a)n}{e} \right)^{1.5(1-2a)n} [3(1 - 2a)n + an/2]^{d/2} (2m)^{3u/2} \prod_{j=1}^{k_4/2} [3(1 - 2a)n + d + 2j - 1] \\
& \times \left[ \left( \frac{3n}{e} \right)^{1.5n} (3n)^{-3u/2} \prod_{i=1}^{d/2+k_1+k_4/2} [3n + (2i - 1)] \right]^{-1} \\
& = 2^{o(n)} a^{3an} [(1 - 2a)n]^{1.5(1-2a)n} \left( \frac{6m}{n} \right)^{3u/2} \prod_{i=1}^{d/2} \frac{3(1 - 2a)n + an/2}{3n + (2i - 1)} \\
& \times \prod_{i=1}^{k_1} \frac{3(w - u) + i}{3n + d + (2i - 1)} \prod_{i=1}^{k_4/2} \frac{3(1 - 2a)n + d + 2i - 1}{3n + d + 2k_1 + 2i - 1} \\
& \leq_o 2^{o(n)} a^{3an} [(1 - 2a)n]^{1.5(1-2a)n} \prod_{i=1}^{d/2} \frac{3(1 - 2a)n + an/2}{3n} \prod_{i=1}^{k_1} \frac{3(w - u) + i}{3n + d + (2i - 1)} \prod_{i=1}^{k_4/2} 1 \\
& \leq_o 2^{o(n)} a^{3an} [(1 - 2a)n]^{1.5(1-2a)n} [(1 - 2a) + a/6]^{d/2} 2^{-k_1} \tag{9}
\end{aligned}$$

Finally we consider the term  $\Pr(d(W) - d(W_1) = d)$  and assume that  $h$  vertices of degree 4 were chosen to be included in  $W \cup W_1$ , so that  $d = h + 3u - 2k_1$ . Then, because there are

$\binom{h}{k_1} \binom{2w-u-h}{(w-u)-k_1}$  out of  $\binom{2w}{w-u}$  ways to distribute the  $k_1$  vertices of degree 4,

$$\begin{aligned} p_3 &= \Pr(d(W) - d(W_1) = d) = \frac{\binom{h}{k_1} \binom{2w-u-h}{(w-u)-k_1}}{\binom{2w-u}{w-u}} \\ &\leq \frac{\binom{h}{k_1} \binom{2w-u-h}{w-u}}{\binom{2w-u}{w-u}} \leq \binom{h}{k_1} \prod_{i=0}^{h-1} \frac{w-i}{2w-i} \\ &\leq 2^{hH(k_1/h)-h} = 2^{k_1} 2^{-k_1+h \cdot H(k_1/h)-h}. \end{aligned}$$

Here  $H(x) = -x \log_2(x) - (1-x) \log_2(1-x)$  is the entropy function. For fixed  $d$  we have  $h = d + 2k_1 + o(n)$ . Thus

$$p_3 \leq 2^{o(n)+k_1+df(k_1/d)}, \text{ where } f(x) = -x + (1+2x)H\left(\frac{x}{1+2x}\right) - (1+2x).$$

$f(x)$  has a unique maximum at  $x^*$ , the solution to  $8x(1+x) = (1+2x)^2$  and  $f(x^*) \leq -0.771$ . Hence

$$p_3 \leq 2^{-0.771d+k_1+o(n)}. \quad (10)$$

Multiplying the bounds in (7), (9), (10) together we have a bound

$$\begin{aligned} p_2 &\leq 2^{o(n)-0.771d+k_1} \left(\frac{1}{a}\right)^{2an} \left(\frac{1}{1-2a}\right)^{(1-2a)n} \left(\frac{4ean}{d}\right)^d \\ &\quad \times a^{3an} (1-2a)^{1.5(1-2a)n} \left(1-2a+\frac{a}{6}\right)^{d/2} 2^{-k_1} \\ &= 2^{o(n)} \left(\frac{2^{1.229}ean}{d}\right)^d a^{an} (1-2a)^{0.5(1-2a)n} \left(1-\frac{11a}{6}\right)^{d/2} \end{aligned}$$

Thus  $p_2 = o(1)$  when  $d = o(n)$ . Let  $d = ban$  for some  $0 < b \leq 1$ . Then,

$$p_2 \leq \left\{ 2^{o(1)} \left[ \frac{2^{1.229}e}{b} \left(1-\frac{11a}{6}\right)^{0.5} \right]^b a(1-2a)^{0.5(1-2a)/a} \right\}^{an}$$

Let  $g(a) = 2^{1.229}e \left(1-\frac{11a}{6}\right)^{0.5}$ . When  $g(a) < e$  then  $\left(\frac{g(a)}{b}\right)^b$  is maximized when  $b = \frac{g(a)}{e}$  which yields

$$p_2 \leq \left\{ 2^{o(1)} e^{2^{1.229}(1-\frac{11a}{6})^{0.5}} a(1-2a)^{0.5(1-2a)/a} \right\}^{an} \leq \left(\frac{99}{100}\right)^{an}.$$

The last inequality is most easily verified numerically.

When  $g(a) > e$  then  $\left(\frac{g(a)}{b}\right)^b$  is maximized at  $b = 1$ . Hence

$$p_2 \leq \left\{ 2^{o(1)} 2^{1.229} e \left(1-\frac{11a}{6}\right)^{0.5} a(1-2a)^{0.5(1-2a)/a} \right\}^{an} \leq \left(\frac{19}{20}\right)^{an}.$$

The last inequality is most easily verified numerically. Thus the probability that there exists a set  $W$  satisfying  $q(V \setminus W) > |W|$  of size  $w = |W| \leq 10^{-5}n$  is bounded by

$$\sum_{w=10^{-5}n}^{0.5n} \left(\frac{99}{100}\right)^w = o(1).$$

This only leaves the case of  $n$  odd. The reader will notice that in none of the calculations above, did we use the fact that  $n$  was even. The Tutte-Berge formula for the maximum size of a matching  $\nu(G)$  is

$$\nu(G) = \min_{W \subseteq V} \frac{1}{2}(|V| + |W| - q(V \setminus W)).$$

We have shown that the above expression is at least  $|V|/2$  for  $W \neq \emptyset$  and so the case of  $n$  odd is handled by putting  $W = \emptyset$  and  $q(W) = 1$ .

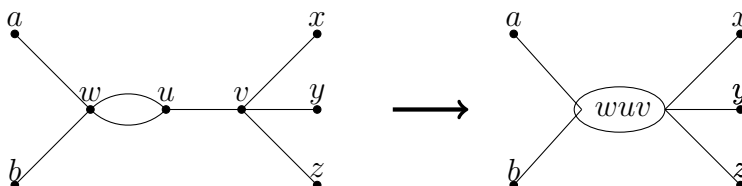
## 5 Conclusions and open questions

The paper of Karp and Sipser [12] has been a springboard for research on matching algorithms in random graphs. Algorithm 1 of that paper has not been the subject of a great deal of analysis, mainly because of the way it disturbs the degree sequences of the graphs that it produces along the way. In this paper we have shown that if the original graph has small maximum degree then the maximum degree is controllable and the great efficiency of the algorithm can be verified.

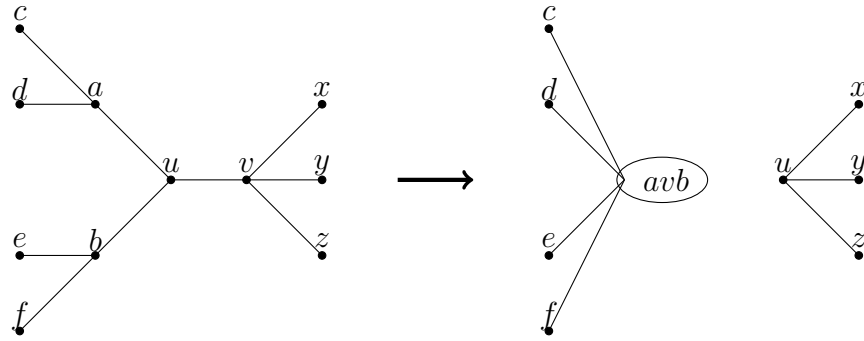
It is natural to try to extend the analysis to random regular graphs with degree more than four and we are in the process of trying to overcome some technical problems. It would also be of interest to analyse the algorithm on  $G_{n,p}$ , as originally intended.

## 6 Diagrams of Hyperactions of interest

**Type 2.**

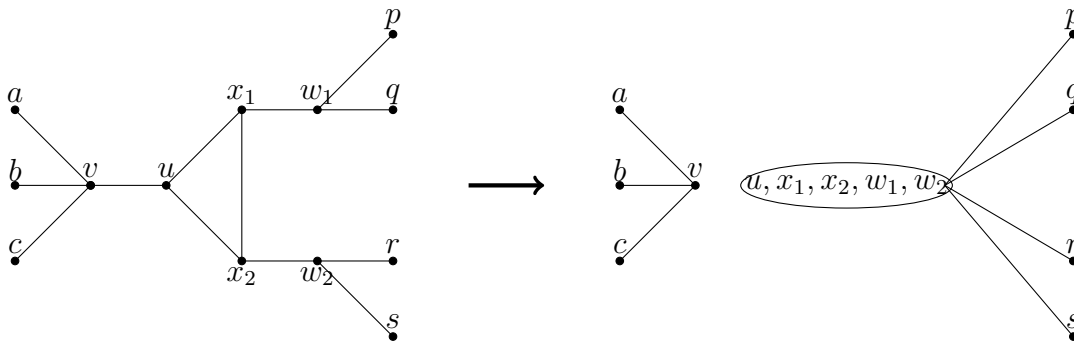


**Type 3.**

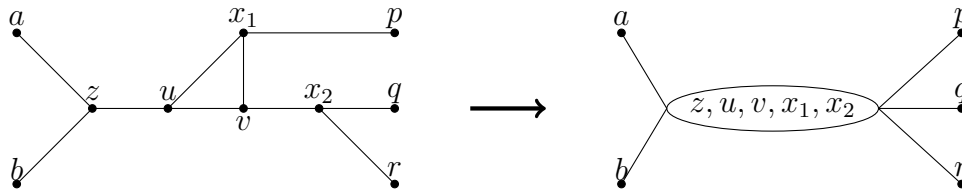


We allow the edge  $\{a, b\}$  to be a single edge in this construction. This gives us a Type 3b hyperaction.

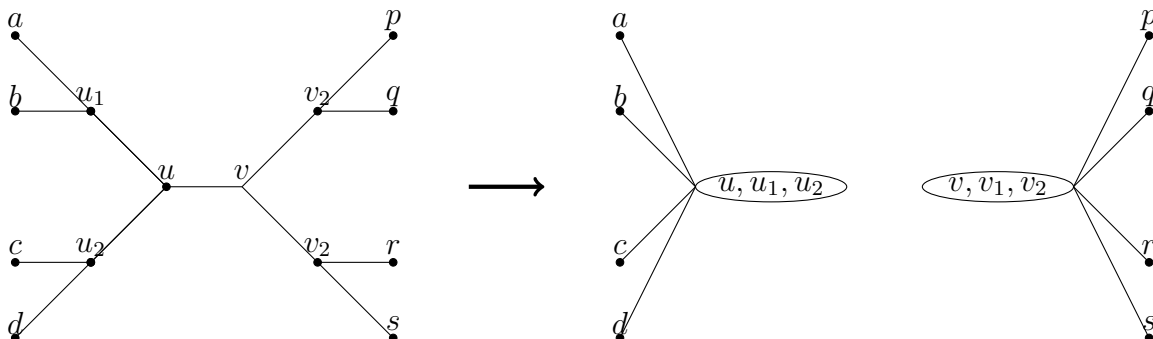
**Type 4.**



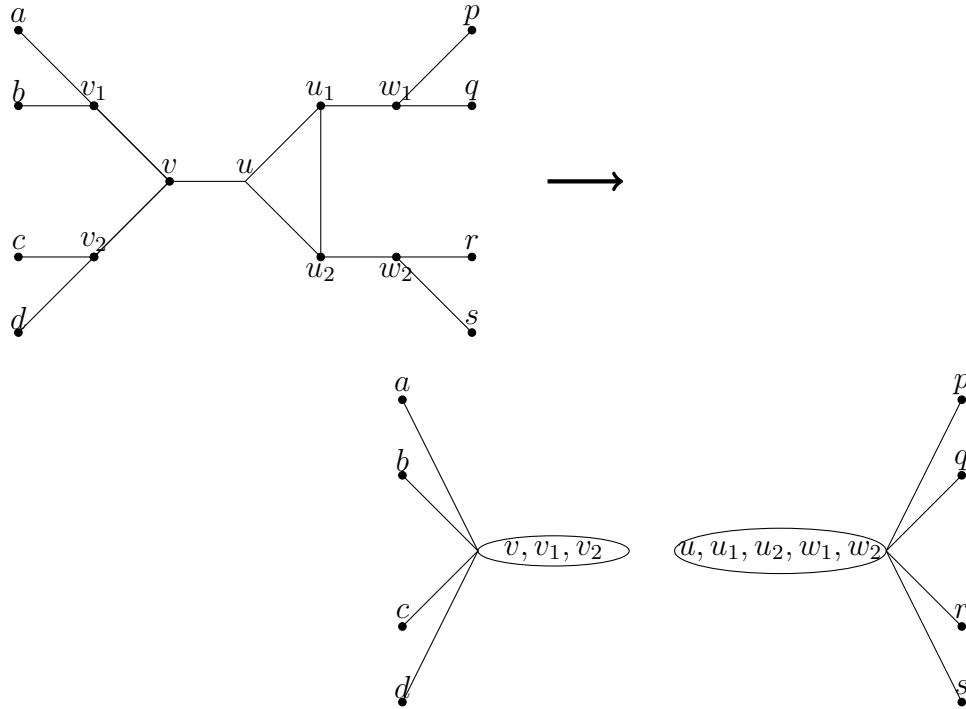
**Type 5.**



**Type 33.**



**Type 34.**



**References**

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

[2] P. Balister and S. Gerke, *Controllability and matchings in random bipartite graphs*, Surveys in Combinatorics 424 (2015) 119-145.

[3] T. Bohman and A.M. Frieze, *Karp-Sipser on random graphs with a fixed degree sequence*, Combinatorics, Probability and Computing 20 (2011) 721-742.

[4] B. Bollobás, *A probabilistic proof of an asymptotic formula for the number of labelled regular graphs*, European Journal on Combinatorics 1 (1980) 311-316.

[5] P. Chebolu, A.M. Frieze and P. Melsted, *Finding a Maximum Matching in a Sparse Random Graph in  $O(n)$  Expected Time*, Journal of the ACM 57 (2010) 1-27.

[6] C. Bordenave and M. Lelarge, *The rank of diluted random graphs*, Annals of Probability 39 (2011) 1097-1121.

[7] A.M. Frieze and M. Karoński, *Introduction to Random Graphs*, Cambridge University Press 2015.

[8] David A. Freedman, *On Tail Probabilities for Martingales*, The Annals of Probability 3 (1975), 100-118.



- [9] A.M. Frieze and B. Pittel, *Perfect matchings in random graphs with prescribed minimal degree*, Trends in Mathematics, Birkhauser Verlag, Basel (2004) 95-132.
- [10] A.M. Frieze, J. Radcliffe and S. Suen, *Analysis of a simple greedy matching algorithm on random cubic graphs*, Combinatorics, Probability and Computing 4 (1995) 47-66.
- [11] W. Hoeffding, W. Hoeffding, Probability inequalities for sums of bounded random variables, *Journal of the American Statistical Association* 58 (1963) 13-30.
- [12] R.M. Karp and M. Sipser, *Maximum matchings in sparse random graphs*, Proceedings of the 22nd Annual IEEE Symposium on Foundations of Computing (1981) 364-375.
- [13] S. Micali and V. Vazirani, *An  $O(V^{1/2}E)$  algorithm for finding maximum matching in general graphs*, 21st Annual Symposium on Foundations of Computer Science, IEEE Computer Society Press, New York. (1980) 1727.