Algorithmic Theory of Random graphs

Alan Frieze* Colin McDiarmid †

August 23, 1996

Abstract

The theory of random graphs has been mainly concerned with *structural properties*, in particular the most likely values of various graph invariants – see Bollobàs [21]. There has been increasing interest in using random graphs as models for the average case analysis of graph algorithms. In this paper we survey some of the results in this area.

1 Introduction

The theory of random graphs as initiated by Erdős and Rényi [52] and developed along with others, has been mainly concerned with *structural properties*, in particular the most likely values of various graph invariantss – see Bollobàs [21]. There has been increasing interest in using random graphs as models for the average case analysis of graph algorithms. We would like in this paper to survey some of the results in this area. We hope to be fairly comprehensive in terms of the areas we tackle and so depth will be sacrificed in favour of breadth.

One attractive feature of average case analysis is that it banishes the pessimism of worst-case analysis. NP-Completeness casts a much smaller shadow. Problems like finding Hamilton cycles may become tractable. Of course one can criticise the models as being unrealistic but they are probably no more so than the pathological examples used in proofs of NP-Completeness and the study of performance guarantees. Furthermore, the models can be close to those used in the empirical testing of algorithms.

1.1 Random Graph Models

Let m, n be natural numbers. The random graph $G_{n,m}$ has vertex set $[n] = \{1, 2, \ldots, n\}$ and edge set $E_{n,m}$, which is a random m-subset of the edges E_n of the complete graph K_n. It is sometimes easier to deal with the independent model $G_{n,p}$, $0 \leq p \leq 1$. This has vertex set $E_{n,p}$ where the edges $e \in E_n$ occur independently in $E_{n,p}$ with probability p. These are the two main models of random graph with which we will be concerned. We shall not discuss geometric models here, but see Steele [189] for a survey.

We will also make a few references to other models of random graphs and to algorithms on random hypergraphs.

*Department of Mathematics, Carnegie Mellon University, Pittsburgh PA15213, U.S.A., Supported by NSF grant CCR-9225008

†Department of Statistics, University of Oxford, Oxford, UK.

1.2 Outline of paper

We divide the paper along the following lines. In Section 2 we discussion algorithms for solving Hamilton cycle and related problems. Section 3 concerns algorithms for solving matching problems.Stable (or independent) set problems are dealt with in Section 4 and graph colouring in Section 5. We then move on to discuss graph isomorphism in Section 6. Section 7 deals with network flows and disjoint path problems. Section 8 discusses shortest paths and minimum spanning trees and Section 9 discusses graph bisection. Section 10 discusses some counting problems. Up to this point the algorithms discussed will all have been sequential in nature. We close with a brief discussion of parallel algorithms that are efficient on average – Section 11.

2 Hamilton Cycles and Related Problems

2.1 Existence

Erdős and Rényi [52] left open the question of the threshold for the existence of a Hamilton cycle in the random graph $G_{n,m}$ i.e. a cycle of length n containing all of the vertices. A breakthrough on this problem came with the paper of Pósa [173] who showed that if $m \geq Kn \log n$, K suitably large, then $G_{n,m}$ is Hamiltonian **whp**[1]. A basic construction of that paper and an earlier one by Komlós and Szemerédi [125] is that of a *rotation*. Here and elsewhere log refers to natural logarithms.

Suppose we have a path $P = x_0, x_1, \ldots, x_k$ in a graph $G = (V, E)$ and we want to find a path of length $k+1$. If x_0 or x_k has a neighbour not in P then one can extend this path by adding a neighbour. Failing this, suppose x_k has a neighbour x_i where $0 \leq i \leq k-2$. If $i = 0$ and G is connected then there is an edge $x_i w$ joining the cycle $x_0, x_1, \ldots, x_k, x_0$ to the rest of the graph and then the path $w, x_i, x_{i+1}, \ldots, x_k, x_0, \ldots, x_{i-1}$ has length $k+1$. We call this a *cycle extension*. If $i \neq 0$ then we perform a *rotation* and construct the path $x_0, x_1, \ldots, x_i, x_k, x_{k-1}, \ldots, x_{i+1}$, which has length k and a *different* endpoint x_{i+1} from which we can iteratively look for further extensions.

Using rotations and extensions Komlós and Szemerédi [126] settled the existence question when they proved

Theorem 1 *If $m = n(\log n + \log\log n + c_n)/2$ then*

$$\lim_{n\to\infty} \mathbf{Pr}(G_{n,m} \text{ is Hamiltonian}) = \begin{cases} 0 & \text{if } c_n \to -\infty \\ e^{-e^{-c}} & \text{if } c_n \to c \\ 1 & \text{if if } c_n \to \infty \end{cases} = \lim_{n\to\infty} \mathbf{Pr}(\delta(G_{n,m}) \geq 2),$$

where δ denotes minimum degree.

Note that throughout the paper log means natural log.

A stronger *hitting time* version of this was proved in Bollobás [23] and also in Ajtai, Komlós and Szemerèdi [1]. Here we add random edges one by one and wait until the first time τ_2 that δ is at least 2.

Theorem 2

$$\lim_{n\to\infty} \mathbf{Pr}(G_{n,\tau_2} \text{ is Hamiltonian}) = 1.$$

[1] A sequence of events $\mathcal{E}_n$ is said to occur **whp** (with high probability) if $\mathbf{Pr}(\mathcal{E}_n) \to 1$ as $n \to \infty$

2.2 Hamilton Cycle Algorithms

2.2.1 General Random graphs

Angluin and Valiant [9] gave a constructive proof of Pósa's result. They described an $O(n(\log n)^2)$ time *randomised* algorithm which **whp** found a Hamilton cycle in $G_{n,m}$, provided $m \geq Kn\log n$, where K is sufficiently large. (For slightly larger K one can use otherwise unexplored parts of the random graph to provide random bits, which makes the algorithm in some sense deterministic). The algorithm consists of a sequence of extensions and rotations which terminate when a Hamilton cycle is found. Shamir [182] improved their result to give a polynomial time algorithm which succeeds **whp** provided $m \geq n(\log n + 3\log\log n)/2$, which is close to being best possible. Finally, Bollobás, Fenner and Frieze [26] described a deterministic algorithm HAM which runs in $O(n^3 \log n)$ time and satisfies

Theorem 3

$$\lim_{n\to\infty} \mathbf{Pr}(HAM \text{ succeeds on } G_{n,\tau_2}) = 1. \tag{1}$$

HAM proceeds in stages. In Stage k there is a path of length k. To proceed to Stage $k+1$ HAM does a set of breadth first searches for a sequence of rotations which will extend the current path.

Although HAM was designed to satisfy (1) it was shown to succeed with probability $1 - o(2^{-n})$ on a random graph $G_{n,1/2}$ chosen uniformly from all graphs with vertex set $[n]$ (success here includes the case of the graph being non-Hamiltonian because it has a vertex of degree 0 or 1.) Thus dynamic programming [94] can be used to take care of the cases where HAM fails to determine whether or not $G_{n,1/2}$ is Hamiltonian, leading to an algorithm which (i) always succeeds, (ii) has polynomial expected time. Gurevich and Shelah [91] described a *linear* expected time algorithm for deciding Hamiltonicity on $G_{n,p}$, for arbitrary *constant* $p > 0$. Their algorithm does not satisfy (1). Subsequently, Thomason [193] described a linear expected time algorithm which is valid for $p \geq n^{-1/3}$.

Research Problem 1 *Find an algorithm that runs in polynomial expected time on $G_{n,m}$ for all values of m.*

Broder, Frieze and Shamir [28] considered the following problem: a graph G is obtained by taking a cycle H of length n and then adding m random edges. Is it possible without knowing H to find a Hamilton cycle in G? If the problem is difficult then the graph G could serve as a *signature*. It would be possible to confirm one's identity by exposing H. The main result of [28] is that if $m \geq c_0 n$ for some absolute constant c_0 then **whp** a Hamilton cycle can be found in G in $O(n^3 \log n)$ time. If instead of adding m random edges, one adds a random perfect matching of $[n]$ then it is still easy to find a Hamilton cycle - Frieze, Jerrum, Molloy, Robinson and Wormald [69].

Research Problem 2 *Suppose we start with a cycle H in which each edge has a different colour from $C = \{c_1, c_2, \ldots, c_n\}$, say. We then add cn random edges and then colour each edge with a random colour from C. Find a polynomial time algorithm which* **whp** *finds a Hamilton cycle with each edge of a different colour.*

2.2.2 Directed Graphs

Rotations cannot be used for digraphs as they reverse the orientation of part of the path. Angluin and Valiant [9] described an algorithm which succeeds **whp** in finding a Hamilton cycle in $D_{n,m}$ provided $m \geq Kn\log n$. At this time the existence question had not been settled: McDiarmid [148]

later showed that $D_{n,m}$ is Hamiltonian **whp** provided $m \geq n(\log n + \log\log n + \omega)$ where $\omega \to \infty$. Subsequently Frieze [64] answered the existence and algorithmic questions. He described an $O(n^{3/2})$ time algorithm DHAM such that

Theorem 4 *If $m = n(\log n + c_n)$ then*

$$\lim_{n\to\infty} \mathbf{Pr}(\textit{ DHAM finds a Hamilton cycle in } D_{n,m}) = \begin{cases} 0 & \textit{if } c_n \to -\infty \\ e^{-2e^{-c}} & \textit{if } c_n \to c \\ 1 & \textit{if } c_n \to \infty \end{cases} \tag{2}$$

$$= \lim_{n\to\infty} \mathbf{Pr}(D_{n,m} \textit{ is Hamiltonian}).$$

Note that the expression on the RHS of (2) is the limiting probability that $D_{n,m}$ has minimal in-degree and out-degree at least 1. Frieze actually proved a (stronger) hitting time version of the above theorem.

2.2.3 Regular graphs

Let $\Omega(r,n)$ denote the set of r-regular graphs with vertex set $[n]$. We consider the case where $r \geq 3$ is fixed and $n \to \infty$. Let $G_{n,r}$ be chosen randomly from $\Omega(r,n)$. It was conjectured for a long time that $G_{n,r}$ is Hamiltonian **whp** for $r \geq 3$. Bollobás [20] and Fenner and Frieze [58] independently showed that $G_{n,r}$ is Hamiltonian **whp** for $r \geq r_0$ (r_0=796 in [58]). Subsequently, Frieze [63] reduced r_0 to 85 by showing that a version of the previously described extension-rotation algorithm succeeded **whp**. Recently Robinson and Wormald [177, 178] solved the existence problem by proving

Theorem 5 *$G_{n,r}$ is Hamiltonian* **whp** *for $r \geq 3$.*

They thus proved the long standing conjecture. Note that G_2 is **whp** a collection of a large number of vertex disjoint cycles. Robinson and Wormald proved more than the existence of one Hamilton cycle. Analysis of their argument shows that **whp** *the number* of Hamilton cycles in $G_{n,r}$ is within an inverse polynomial factor ($p(n)^{-1}$) of the number of 2-factors of $G_{n,r}$. Exploiting this, Frieze, Jerrum, Molloy, Robinson and Wormald [69] came up with the following simple algorithm: Use the algorithm of Jerrum and Sinclair [99] to generate (near) random 2-factors of $G_{n,r}$ until a Hamilton cycle is produced. The expected number of 2-factors produced before a Hamilton cycle is found is then $O(p(n))$.

Research Problem 3 *Extend the results of [69] on finding a Hamilton cycle in $G_{n,r}$ to the case where r grows with n.*

2.3 Travelling Salesman Problems

In this section we discuss travelling salesman problems (TSP's) where the edge weights are drawn *independently* from the same (non-negative) distribution.

2.3.1 Asymmetric TSP's and the Assignment problem

The *Assignment Problem (AP)* is the problem of finding a minimum-weight perfect matching in an edge-weighted bipartite graph. An instance of the AP can be specified by an $n \times n$ matrix $M = (m_{i,j})$; here $m_{i,j}$ represents the weight of the edge between x_i and y_j, where $X = \{x_1, x_2, \ldots, x_n\}$ is the set of "left vertices" in the bipartite graph, and $Y = \{y_1, y_2, \ldots, y_n\}$ is the set of "right vertices".

The AP can be stated in terms of the matrix M as follows: find a permutation σ of $\{1,2,\ldots,n\}$ that minimizes $\sum_{i=1}^{n} m_{i,\sigma(i)}$. Let $AP(M)$ be the optimal value of the instance of the AP specified by M.

The *Asymmetric Traveling-Salesman Problem (ATSP)* is the problem of finding a Hamiltonian circuit of minimum weight in an edge-weighted directed graph. An instance of the ATSP can be specified by an $n \times n$ matrix $M = (m_{i,j})$ in which $m_{i,j}$ denotes the weight of edge (i,j). The ATSP can be stated in terms of the matrix M as follows: find a cyclic permutation π of $\{1,2,\ldots,n\}$ that minimizes $\sum_{i=1}^{n} m_{i,\pi(i)}$; here a cyclic permutation is one whose cycle structure consists of a single cycle. Let $ATSP(M)$ be the optimal value of the instance of the ATSP specified by M.

It is evident from the above two definitions that $AP(M) \leq ATSP(M)$. The ATSP is NP-hard, whereas the AP is solvable in time $O(n^3)$.

Karp [108] considered the case where the $m_{i,j}$ are drawn independently from the uniform distribution over [0,1]. He proved that a certain *Patching Algorithm* produced a near optimal solution **whp**. In fact he showed that the tour produced by this algorithm was good enough to yield

Theorem 6

$$ATSP(M)/AP(M) = 1 - o(1) \quad \textbf{whp}. \tag{3}$$

The approach is as follows:

- Solve the problem $AP(M)$.
- **Whp** there are less than $2\log n$ cycles in the optimum solution. This is because the optimum solution will correspond to a random permutation of $[n]$.
- Patch the cycles one by one into the current largest cycle, as cheaply as possible. Patching two cycles involves deleting an edge from each and adding two edges to produce one large cycle.

Later, Karp and Steele [118] and then Dyer and Frieze [47] strengthened this result in several ways. For example the latter paper shows that the error term in (3) is $o((\log n)^4/n)$.

Research Problem 4 *Determine the correct asymptotics for this error term.*

Research Problem 5 *The assignment problem is often used as a lower bound in branch and bound algorithms for solving the asymmetric travelling salesman problem. Assuming costs are iid uniform on [0,1], determine if there is a version of branch and bound which runs in polynomial time* **whp**.

Research Problem 6 *Determine* $\lim_{n\to\infty} \mathbf{E}(AP(M))$ *when the entries of* M *are iid uniform on [0,1]. This limit exists – Aldous [4], Avram and Bertsimas [12] – and is conjectured to be* $\zeta(2) = \pi^2/6$. *It is known to be at least* 1.44 *– Goemans and Kodialam [83]) – and at most 2 – Karp [111], Dyer Frieze and McDiarmid [49].*

More recently, inspired by the computational results of Miller and Pekny [159], Frieze, Karp and Reed [70] considered distributions for the $m_{i,j}$ where there is a significant chance that $m_{i,j} = 0$. They proved

Theorem 7 *(a) Let* (X_n) *be a sequence of non-negative random variables. Let* $p_n = Pr[X_n = 0]$ *and let* $w(n) = np_n$. *Let* $M = M(n)$ *be an* $n \times n$ *matrix whose entries are drawn independently from the same distribution as* X_n. *If* $w(n) \to \infty$ *as* $n \to \infty$ *then* $AP(M) = ATSP(M)$ **whp**.

(b) Let $M = M(n)$ be an $n \times n$ matrix whose entries are drawn independently from the uniform distribution over $\{0, 1, ..., \lfloor cn \rfloor\}$ where c is a positive constant. Then, the probability that $AP(M) \neq ATSP(M)$ does not tend to zero as n tends to infinity.

(c) Let $M = M(n)$ be an $n \times n$ matrix whose entries are drawn independently from the uniform distribution over $\{0, 1, ..., \lfloor c_n n \rfloor\}$ where c_n tends to infinity with n. Then, the probability that $AP(M) \neq ATSP(M)$ tends to 1 as n tends to infinity.

The proof of Theorem 7(a) relies on an algorithm which converts the assignment solution to a tour only using edges of length zero. Preliminary to a patching phase there is a phase which increases the minimum cycle size so that patching is likely to be successful.

Research Problem 7 *Determine the limiting probability that $AP(M) = ATSP(M)$ in case (b) of Theorem 7.*

2.3.2 Symmetric TSP's

There has been less success in finding algorithms which (nearly) solve symmetric TSP's ($m_{i,j} = m_{j,i}$) **whp**. This is probably because the natural analogue of the assignment problem relaxation, the minimum weight 2-factor problem lacks one particular property. The optimum solution to a random problem is not a random 2-factor.

Indeed the analogue of Karp's early result (3) for uniform [0,1] costs has not been proved with assignments replaced by 2-factors. There is an earlier result of Frieze [62] which is weaker than Theorem 7(a).

Theorem 8 *Let $M = M(n)$ be an $n \times n$ symmetric matrix whose entries are drawn independently from the uniform distribution over $\{0, 1, ..., \lfloor n/(\omega \log\log n) \rfloor\}$ where $\omega \to \infty$. There is an $O(n^3 \log n)$ time algorithm which solves the corresponding TSP* exactly*,* **whp**.

The idea here is to convert the problem into a constrained Hamilton cycle problem. Vertices incident with few zero length edges are optimally covered with vertex disjoint paths. Then a Hamilton cycle is found which contains these paths as sub-paths and otherwise uses zero length edges. This necessarily solves the TSP exactly.

Research Problem 8 *Remove the* $\log\log n$ *factor in Theorem 8.*

3 Matchings

3.1 Existence

The threshold for the existence of a perfect matching in $G_{n,m}$ was found by Erdős and Rényi [53] in their early sequence of papers. They proved

Theorem 9 *If $m = n(\log n + c_n)/2$ then*

$$\begin{aligned} \lim_{n\to\infty} \mathbf{Pr}(G_{n,m} \text{ has a perfect matching}) &= \begin{cases} 0 & \text{if } c_n \to -\infty \\ e^{-e^{-c}} & \text{if } c_n \to c \\ 1 & \text{if } c_n \to \infty \end{cases} \\ &= \lim_{n\to\infty} \mathbf{Pr}(\delta(G_{n,m}) \geq 1). \end{aligned}$$

Essentially the same result is true for bipartite graphs, Erdős and Rényi [54]. There are also corresponding hitting time versions, as in Theorems 1 and 2.

The existence of perfect matchings has been studied in other models too. It follows from Tutte's theorem [199] on perfect matchings that for $r \geq 3$, an (r-1)-edge connected r-regular graph (with rn even) has a perfect matching. It follows that a random r-regular graph, $r \geq 3$ fixed, has a perfect matching **whp**, Bollobás [21].

Walkup [202] considered bipartite graphs with n vertices in each part in which each vertex independently chooses k random neighbours. He showed that for fixed $k \geq 2$ such a graph has perfect matching **whp**.

3.2 Algorithms for finding perfect matchings

The basic tool for finding a perfect matching is the augmenting path. Angluin and Valiant [9] gave an $O(n \log n)$ expected time algorithm that finds a perfect matching in $G_{n,m}$ **whp**, provided $m \geq Kn \log n$ for sufficiently large K. It basically relies on (randomised) depth first search to find augmenting paths. Goldschmidt and Hochbaum [84] gave an algorithm which improves the running time to $O(n \log(n^2/m))$ but still for $m \geq Kn \log n$. The above algorithms are designed to run on random graphs. Motwani [163] analysed the performance of some standard worst-case efficient algorithms on random graphs and showed that their expected running times are significantly better than the worst-case suggests. For example, the algorithm of Micali and Vazirani [158] has worst-case time complexity $O(n^{1/2}m)$ on a graph with n vertices and m edges, but on $G_{n,m}$, with $2m/n - \log n \to \infty$ it terminates in $O(m \log n / \log \log n)$ time **whp**. The main reason being that **whp**

$$\text{every non-perfect matching has an augmenting path of length } O(\log n / \log d), \tag{4}$$

where $d = 2m/n$ is the average degree.

Motwani applied his analysis to Walkup's model as well.

It is worth remarking that (4) and the fact that for m considered here **whp** $\Delta(G_{n,m}) = O(d)$ together imply that the ratio of the number of near perfect to perfect matchings is bounded by a fixed polynomial in n. (A matching is *near perfect* if it covers all but 2 vertices.) This means that **whp** a certain randomised algorithm (Jerrum and Sinclair [99]) can be used to approximate the number of perfect matchings in almost all graphs. ([99] did not give a tightest possible result for random graphs and Motwani's paper filled in a small gap.)

3.3 Sparse Random graphs

If $m = \lfloor cn \rfloor$, $c > 0$ constant then $G_{n,m}$ has no perfect matching **whp**, see Theorem 9. Indeed there are a large number of isolated vertices. For such a *sparse* random graph the interest is in using a simple heuristic to find a large matching which is close to optimal **whp**. Researchers have concentrated in the main on the analysis of greedy heuristics:

GREEDY

begin
  $M \leftarrow \emptyset$;
  while $E(G) \neq \emptyset$ **do**
  begin
    A: Choose $e = \{u, v\} \in E$
    $G \leftarrow G \setminus \{u, v\}$;
    $M \leftarrow M \cup \{e\}$

end;
Output M
end

($G \setminus \{u,v\}$ is the graph obtained from G by deleting the vertices u, v and all edges incident with them, together with any vertices which become exposed.)

The average performance of GREEDY when the input is random was first analysed by Tinhofer [193]. He considered its performance on the random graph $G_{n,p}$ in the dense case where p is fixed independent of n. In this case it is fairly easy to show that the algorithm produces a matching of size $n/2 - O(\log n)$ **whp**.

Let $X = X(n,m)$ be the random number of edges in the matching produced by GREEDY applied to $G_{n,m}$ when the edge choice in statement **A** is uniformly random. Dyer, Frieze and Pittel [50] were able to establish the asymptotic distribution of this variable when $m = \lfloor cn \rfloor$. Let

$$\begin{aligned} \phi(c) &= \frac{c}{2(c+1)} \\ \psi(c) &= \frac{c^2(c+3)}{6(c+1)^4}. \end{aligned}$$

Theorem 10 *As $n \to \infty$ $(X(n,m) - n\phi(c))/\sqrt{n\psi(c)}$ converges in distribution, and with all its moments, to the standard normal variable with mean zero and variance one.*

Thus $X(n,m)$ is asymptotically Gaussian with mean $n\phi(c)$ and variance $n\psi(c)$. As one should expect, $\phi(c) \to 1/2$ as $c \to \infty$ which corresponds to a matching of size $n/2 - o(n)$.

It is possible to modify this algorithm without considerable complications, so as to improve its likely performance. Perhaps the simplest modification is to first choose a vertex v at random and then to randomly choose an edge incident with v. We refer to this as MODIFIED GREEDY. Dyer, Frieze and Pittel also analysed the performance of MODIFIED GREEDY in the same setting as for GREEDY. Let $\hat{X} = \hat{X}(n,m)$ be the random number of edges in the matching produced by MODIFIED GREEDY on $G_{n,m}$. Let

$$\hat{\phi}(c) = \tfrac{1}{2} - \frac{\log(2 - e^{-c})}{2c} > \phi(c).$$

Theorem 11 *As $n \to \infty$ $(\hat{X}(n,m) - n\hat{\phi}(c))/\sqrt{n\hat{\psi}(c)}$ converges in distribution, and with all its moments, to the standard normal variable with mean zero and variance one. (Here $\hat{\psi}(c)$ is the unknown solution of a certain differential equation.)*

MODIFIED GREEDY was also discussed in Tinhofer [193] as well as by Goldschmidt and Hochbaum [84] who proved probabilistic lower bounds on the size of the matching produced in $G_{n,p}$. In particular Goldschmidt and Hochbaum prove a probabilistic lower bound of $n(1 - (1+\epsilon)/c)/2$ for any fixed $\epsilon > 0$, (which is smaller than $\phi(c)$).

GREEDY and MODIFIED-GREEDY both find matchings which are less than the maximum by a constant factor. Karp and Sipser [117] considered a similar greedy type of algorithm which we will call KSGREEDY. Their algorithm (a) chooses an edge incident to a vertex of degree 1 while there is one and otherwise (b) chooses a random edge. The algorithmic change is tiny, but the improvement in performance is spectacular. They show that this algorithm is asymptotically optimal in the sense that with high probability it finds a matching which is within $o(n)$ of the optimum size! They also prove that if $c \leq e$ then KSGREEDY spends almost all of its time in case (a). The algorithm is considered to run in two phases. Phase 1 ends when the minimum degree of the graph that remains

is at least two. Note that during Phase 1 the algorithm makes correct choices in the sense that the edges chosen are a subset of some maximum matching.

Aronson, Frieze and Pittel [10] have been conducting a further analysis of this algorithm. At this stage it looks as though the following is true **whp**:

- If $c < e$ then at the end of Phase 1, all that is left of the graph is a few vertex disjoint cycles.
- If $c > e$ then in Phase 2, KSGREEDY will match all but about $n^{1/5}$ of those vertices which remain at the end of Phase 1. More precisely, there exist positive constants c_1, c_2, a such that if L denotes the number of vertices which become isolated in Phase 2, then

$$c_1 n^{1/5} \leq \mathbf{E}(L) \leq c_2 n^{1/5} (\log n)^a. \tag{5}$$

  Here a is likely to be 1 or 2.
- Analysis of the algorithm gives an asymptotic expression for the size of the maximum matching in $G_{n,m}$.

(Bollobás and Brightwell [24] did some analysis of this algorithm applied to a random bipartite graph with at most en random edges).

Research Problem 9 *Find the correct exponent of* $\log n$ *in (5).*

Research Problem 10 *Determine the asymptotics for the difference between the number of vertices left isolated by KSGREEDY and the number left isolated by a maximum matching in a sparse random graph.*

Another possible version of GREEDY is MINGREEDY where in Step A one chooses a (random) vertex of minimum degree and then a random neighbour of this vertex. Frieze, Radcliffe and Suen [75] considered the performance of MINGREEDY on random cubic graphs. They proved

Theorem 12 *Let L_n denote the number of vertices left exposed by the matching constructed by running MINGREEDY on a random cubic graph with n vertices. Then there exist constants $d_1, d_2 > 0$ such that*

$$d_1 n^{1/5} \leq \mathbf{E}(L_n) \leq d_2 n^{1/5} \log n. \tag{6}$$

(Recall that a random cubic graph has a perfect matching **whp**.) Thus MINGREEDY usually does very well. Note the common exponent 1/5 in (5) and (6). This can be explained to some extent by the fact that near the end of KSGREEDY, when most *avoidable* vertex isolations are made, the maximum degree is bounded **whp**.

In computational experiments MINGREEDY left an average of just over 10 vertices unmatched when run on random cubic graphs with 10^6 vertices.

There are some technical difficulties involved in extending the analysis to random r-regular graphs, $r \geq 4$.

Research Problem 11 *Analyse the performance of MINGREEDY on $G_{n,r}$ for $r > 3$.*

Karp, Rinnooy-Kan and Vohra [116] gave an algorithmic version of Walkup's result on perfect matchings in random bipartite graphs [202] and used for an algorithmic version of his result on the assignment problem [203] – see also Avis and Lai [11] for a slightly weaker result in this vein.

Suppose the vertices of our bipartite graph are partitioned into sets A, B. Each $a \in A$ makes two random choices in B. Let e_a denote this pair of choices and let $G_B = (B, \{e_a : a \in A\})$. Define f_b and the graph $G_A = (A, \{f_b : b \in B\})$ similarly. Note that both G_A and G_B are close in distribution to $G_{n,n}$. Initially set $H_1 = G_A$ and $H_2 = (B, \emptyset)$. Consider an isolated tree T of H_1, (if there is one). Choose any vertex x as root. Orient the edges of T towards x. Let the directed edges of T now be $f_{b_i} = (a_i, a_i')$ for $1 \le i \le t$. If we match (a_i, b_i) together for $1 \le i \le t$ then only the root x is not matched. So what we do is consider matching it with one of b, b' where $e_x = \{b, b'\}$. But if we say match x with b then, to avoid later conflicts we should delete f_b from H_1. We thus go to and fro between H_1 and H_2, rooting trees in H_1, adding edges to H_2 and deleting edges in H_1. If successful, the algorithm transforms H_1 and H_2 into graphs with every component either a tree or unicyclic, such that a perfect matching can be constructed from the components of the two graphs. The main result of the paper is

Theorem 13 Whp *the above,* Pairing *algorithm, terminates with a perfect matching.*

Cooper and Frieze [40] used this constructive proof of Walkup's result in proving that the random digraph $D_{2-in,2-out}$ is Hamiltonian **whp**. This is a pleasant symbiosis between the algorithmic and structural theory of random graphs.

Research Problem 12 *Frieze [59] gave an extension to non-bipartite graphs of the result of [202]on the existence of matchings in bipartite graphs. Can the pairng algorithm of [116] discussed above also be extended to non-bipartite graphs?*

3.4 Related matching problems

Knuth, Motwani and Pittel [122], Pittel [167, 168, 169] studied structural properties of random instances of *Stable Marriage* – see Gale and Shapley [81]. Pittel [170] did some analysis of the proposal algorithm for the *Stable Roommates* Problem.

Research Problem 13 *Compute the expected number of rounds in the Gale-Shapley proposal algorithm for the stable marriage problem, assuming that preferences are random and independent.*

Research Problem 14 *If preferences in an instance of the stable roomates problem are random and independent, determine the limiting probability that there is a stable solution.*

Frieze and Pittel [74] analysed a similar algorithm for a similar matching problem, viz. the Shapley-Scarfe [187] algorithm for finding a core allocation in a market with indivisible goods.

4 Stable (or Independent) sets

A *Stable* (or *independent*) set in a graph G is a set of vertices no two of which are adjacent; and the *stability* (or independence) number $\alpha(G)$ is the maximum size of a stable set.

4.1 Random graphs $G_{n,p}$

The behaviour of the stability number $\alpha(G_{n,p})$ is well understood, at least when the average degree is large. We assume here that the edge-probability $p = 1 - q$ is bounded below 1.

Theorem 14 *If $np \to \infty$ as $n \to \infty$, then $\alpha_n = \alpha(G_{n,p})$ satisfies*

$$\alpha_n \sim 2\log np/\log 1/q \quad \textbf{whp}.$$

(Note that if $p \to 0$ then $\log(1/q) \sim p$.) When p is a constant, α_n is in fact remarkably concentrated: there exists $k = k(n)$ such that $\alpha_n = k$ or $k+1$ **whp**. These results are proved by the (non-algorithmic) second moment method for p constant – see [21], together with a concentration inequality for $p \to 0$ – see Frieze [65].

The *greedy* or *sequential* stable set algorithm picks a stable set in a graph G by looking at the vertices in some fixed order, and adding a vertex to the current stable set whenever possible. Let $\sigma(G)$ be the size of the stable set produced. Observe that if t vertices remain after adding the kth vertex to the stable set, then independently of any previous history, the probability that we need to look at more than i more vertices to pick up the $k+1$st vertex equals $(1-q^k)^i$ for each $i \le t$. This shows that we can analyse $\sigma_n = \sigma(G_{n,p})$ by considering sums of independent geometrically distributed random variables: see Grimmett and McDiarmid [88], Fernandez de la Vega [56], Gazmuri [82] and McDiarmid [150].

Theorem 15 *If $np \to \infty$ as $n \to \infty$ then $\sigma_n \sim \log np/\log \frac{1}{q}$ **whp**; and if $np \to c > 0$ as $n \to \infty$ then $\sigma_n \sim n\log(1+c)/c$ **whp**.*

Now suppose that $np \to \infty$ as $n \to \infty$. Then we have $\sigma_n/\alpha_n \sim \frac{1}{2}$ **whp**. Is this good? Let $\hat{\alpha}(G)$ be the minimum size of a maximal stable set in G. Then $\hat{\alpha}_n/\alpha_n \sim \frac{1}{2}$ **whp**, which does not sound good for the greedy algorithm. However, it is not known if for some constant $\delta > 0$ there is a polynomial time algorithm which yields a stable set of size at least $(\frac{1}{2}+\delta)\alpha_n$ with say probability at least $\frac{1}{2}$. For related questions see Karp [106], McDiarmid [150] and Jerrum [98].

Research Problem 15 *Construct a polynomial time algorithm that finds an independent of size at least $(\frac{1}{2}+\delta)\alpha_n$ **whp** or show that such an algorithm does not exist modulo some reasonable conjecture in the theory of computational complexity such as e.g. $P \neq NP$.*

For corresponding results for random hypergraphs, see [82] and the papers mentioned in Section 5.5 below.

4.2 Random regular graphs $G_{n,r}$

When r is large the behaviour of $\alpha(G_{n,r})$ is like that for $\alpha(G_{n,p})$ with $np = r$, see Frieze and Łuczak [72]. Thus the value $\alpha(G_{n,r})$ is fairly well pinned down (ultimately from a corresponding result about $\alpha(G_{n,p})$) but there is no polynomial time algorithm known to obtain stable sets of about this size.

Theorem 16 *Let $f(r) = \frac{2}{r}(\log r + \log\log r + 1 - \log 2)$. For any constant $\epsilon > 0$ there exists a constant r_ϵ such that if $r_\epsilon \le r = r(n) \le n^{\frac{1}{3}}$ then*

$$\left|\frac{\alpha(G_{n,r})}{n} - f(r)\right| \le \frac{\epsilon}{r} \quad \textbf{whp}.$$

The above theorem says nothing about small values of r, for example $r = 3$. Here things are rather different. The behaviour of $\alpha(G_r)$ is not well determined, and the best known lower bound comes from analysing the 'mingreedy' algorithm. This algorithm repeatedly deletes from the graph a vertex v of minimum degree, and if possible adds v to the current stable set in which case it also

deletes all the neighbours of v. Let $\sigma^{min}(G)$ denote the size of the stable set found (with ties broken randomly). For cubic graphs, we find Frieze and Suen [79], Wormald [205] that $\frac{1}{n}\sigma(G_{n,3}) \sim 0.375$ **whp** and $\frac{1}{n}\sigma^{min}(G_{n,3}) \sim 0.4328..$ **whp**. The best upper bound known is $\frac{1}{n}\alpha(G_{n,3}) \leq 0.4534$ **whp**. The results on the algorithms are proved in [205] by a general approach which relates the random processes to solutions of certain differential equations.

Research Problem 16 *Are the results of [79], [205] optimal i.e. does the algorithm mingreedy find an asymptotically maximal independent set?*

4.3 Difficulty results

Suppose that we wish to prove upper bounds on the stability number of a graph $G = (V, E)$. Chvátal [38] formalises certain natural (and powerful) rules that we might use.

A *statement* is a pair (S, t) where $S \subseteq V$ and t is a non-negative integer – to be interpreted as $\alpha(S) \leq t$ where $\alpha(S)$ is the size of the largest stable subset of S. A *recursive proof* of a statement (S, t) is a sequence of statements (S_i, t_i) $i = 0, 1, \ldots, m$ where $(S_0, t_0) = (\emptyset, 0)$, $(S_m, t_m) = (S, t)$, and such that each statement (S_k, t_k) for $k \geq 1$ can be deduced from the earlier statements by either the dichotomy rule or the monotone rule.
The dichotomy rule: from $(S \setminus \{v\}, x)$ and $(S \setminus (\{v\} \cup N(v)), y)$, we can deduce $(S, \max(x, y + 1))$. Here $N(v)$ denotes the set of neighbours of v.
The monotone rule: from (S, x) we can derive (S', x') when $S' \subseteq S$ and $x' \geq x$.

If there is a recursive proof of (S, t) then $\alpha(S) \leq t$, and conversely. However, for almost all graphs $G = (V, E)$ with a (sufficiently large) linear number of edges, all recursive proofs of $(V, \alpha(G))$ must have at least exponential length [38]. For related results see Pittel [166].

Let us return to the problem of finding a stable set of size at least $(1 + \delta) \log_2 n$ in $G_{n,p}$ where $p = \frac{1}{2}$. Jerrum [98] investigated how well a natural Metropolis algorithm will handle the (complementary clique) problem.

A parameter $\lambda > 1$ is fixed (corresponding to inverse temperature). If the current state is the stable set S then the next stable set S' is obtained as follows. A vertex v is picked uniformly at random. There are three cases: (i) if $v \in S$ then with probability λ^{-1} let $S' = S \setminus \{v\}$, else let $S' = S$; (ii) if $v \notin S$ and $S \cup \{v\}$ is stable then let $S' = S \cup \{v\}$; and (iii) otherwise let $S' = S$.

It is shown in [98] that if $\lambda \sim n$ then the desired large stable sets are favoured at equilibrium; but that, whatever the value of λ, for some initial state the expected time to reach a large stable set grows super-polynomially. A similar result holds even if a stable set of size n^β where $0 < \beta < \frac{1}{2}$ is 'hidden' in the random graph. See also Kučera [131].

5 Colouring

A (vertex) colouring of a graph G is an assignment of colours to the vertices of G such that adjacent vertices receive different colours. The *chromatic number* $\chi(G)$ is the least number of colours possible.

It has long been known that the problem of testing if $\chi(G) \leq k$ is NP-complete, even for any fixed $k \geq 3$. Recently it has been shown that if P $\neq$ NP then it is not possible to approximate the chromatic number within a ratio $O(n^\epsilon)$ for some constant $\epsilon > 0$ Lund and Yannakakis [143]; though it *is* possible to do so within a ratio $\tilde{O}(n^{\frac{1}{4}})$ Karger, Motwani and Sudan [103] (i.e. ignoring some log factors). Also, it is NP-hard to 4-colour an arbitrary 3-colourable graph Khanna, Linial and Safra [120]. It is thus natural to consider algorithms that usually colour well.

5.1 Random graphs $G_{n,p}$

Dense case
We first consider the case when the edge probability p is constant. The behaviour of the chromatic number $\chi_n = \chi(G_{n,p})$ is well understood Bollobás [22] (see also Kučera and Matula [134]).

Theorem 17 *Let p be fixed. Then $\chi(G_{n,p}) \sim n/\alpha_n$* **whp**.

Indeed, we can be more precise [153] : for the function f mentioned in Theorem 16 we have $\chi(G_{n,p}) = n/(\alpha_n + O(1))$ **whp**. The lower bound here (and later in this section) comes from the inequality $\chi(G)\alpha(G) \geq |V|$ and what we already know about stable sets. The proof of the upper bound in [22], subsequently refined in McDiarmid [153], uses concentration inequalities to show that **whp** in $G = G_{n,p}$ every large set of vertices contains a stable set of size nearly $\alpha(G)$. Thus we can colour nearly optimally by repeatedly stripping off maximum sized stable sets.

The *greedy* or sequential colouring algorithm considers the vertices in a fixed order, and gives each vertex the first available colour. This is equivalent to repeatedly applying the greedy stable set algorithm to strip off colour sets. Let $\gamma(G)$ be the number of colours used, and use γ_n to denote $\gamma(G_{n,p})$. The behaviour of γ_n was known long before that of χ_n, see Grimmett and McDiarmid [88] and also Bollobás and Erdős [25].

Theorem 18 *Let p be fixed. Then $\gamma_n \sim 2\chi_n$* **whp**.

This result may be proved from our knowledge of χ_n and the greedy stable set algorithm. Does this theorem show that the greedy colouring algorithm is good? It turns out that several variants of this approach, for example picking a polynomial number of different random vertex orders, use about the same number of colours – see McDiarmid [147] and also McDiarmid [149] and Kučera [130]. It is not known if any polynomial time algorithm uses at most $(2-\delta)\chi_n$ colours with probability at least $\frac{1}{2}$ say, for some $0 < \delta \leq 1$.

Research Problem 17 *Is there such an algorithm? (See also Problem 15).*

Matula [145] gave a (super-polynomial time) algorithm involving local searches for large stable sets and the expose-and-merge idea, which uses $\leq (\frac{4}{3} + \delta)\chi_n$ colours **whp**. (This gave the best upper bound known at the time on χ_n.)

Sparse case
Next let us consider $G_{n,p}$ with $p = p(n) = o(1)$. As long as $np \to \infty$ things are much as in the case p constant, with $\log \frac{1}{q}$ replaced by p. Łuczak [140] extended the proof of Kučera and Matula [134] to show that $\chi_n \sim np/2\log np$ **whp**. For sparse graphs the greedy algorithm needs a tidy-up end phase. Suppose that we have repeatedly stripped off colour sets until the average degree has fallen to about 1. Then the remaining graph has at most one cycle per component with high probability and may then be easily 3-coloured. Let γ'_n be the number of colours used on $G_{n,p}$ by a corresponding modified greedy algorithm. Then Shamir and Upfal [185] and Fernandez de la Vega [56] show $\gamma'_n \sim 2\chi_n$ **whp**. Recently, Pittel and Weishar [171] have analysed greedy colouring without the tidy up phase. They prove that **whp** the number of colours used grows as $\log_2 \log_e n$. The interest in this result stems from viewing it as colouring a random graph which is presented and has to be coloured vertex by vertex on-line.

When $np \sim c$ for some small constant c we may still analyse γ'_n McDiarmid [150], but things are less clear for the chromatic number Łuczak and Wierman [141], Chvátal [39], Molloy and Reed [162], Molloy [161]. In this context the notion of k-core becomes important. The k-core of a graph being the largest subgraph of minimum degree at least k. A graph without a k-core has chromatic number

at most k – remove a vertex of degree at most $k-1$ and apply induction. It is therefore important to know the thresholds for the existence of k-cores. This was done by Pittel, Spencer and Wormald [172] who analysed a simple algorithm for finding cores.

Very dense case
Usually we assume that p is bounded below 1, but suppose that $p \to 1$ in such a way that the largest stable set stays bounded. In particular let r be a fixed positive integer and suppose that $qn^{2/r} \to \infty$ and $qn^{2/(r+1)} \to 0$. We find in Fernandez de la Vega [57] that **whp** $\alpha_n \leq r+1$ and there are $o(n)$ stable sets of size $r+1$ so that $\chi_n \geq (1-o(1))n/r$: however, an arbitrary choice of stable r-sets leads to a polynomial time algorithm which uses $\leq (1+o(1))n/r$ colours [57].

5.2 Random regular graphs $G_{n,r}$

As for the stability number, when r is large the situation with $\chi(G_{n,r})$ is like that for $\chi(G_{n,p})$ – see Frieze and Łuczak [72]. Little work seems to have been done on the case of r small (though it is easy to see that $\chi(G_{n,3}) = 3$ **whp**).

5.3 Random k-colourable graphs

There has been much interest in colouring random k-colourable graphs, where k is fixed or slowly growing with the number n of vertices, and in particular in the case $k = 3$. Several probabilistic models have been considered.

Uniform models
Perhaps the most natural probabilistic model is the *uniform k-colourable model*, when we sample uniformly from all k-colourable graphs on the vertices $1, \ldots, n$ — see Turner [198], Dyer and Frieze [46]. [46] gives a three stage algorithm which, for fixed k, k-colours graphs in the uniform k-colourable model in polynomial expected time. The first stage uses the mingreedy stable set algorithm as in Kučera [127] to strip off $k-1$ colour sets. With high probability this leaves a stable set and thus succeeds in giving a k-colouring. If this stage fails, then repeatedly partial colourings are tested to see if they lead to a nearly 'forced' k-colouring (all but a few vertices have exactly one colour available). This second stage is very unlikely to fail, but if it does then all possible k-colourings are tried.

A different uniform model was proposed in Prömel and Steger [174]: here graphs are drawn uniformly from the set of all graphs on vertex set $[n]$ with no subgraph the complete graph K_{k+1} on $(k+1)$ vertices, where k is fixed. Such graphs are **whp** k-colourable – see Kolaitis, Prömel and Rothschild [123]. [174] gives an algorithm which colours such graphs optimally in $O(n^2)$ expected time (and which thus performs similarly in the uniform k-colourable model). The first stage of their algorithm fails to find a forced k-colouring with exponentially small probability. The notion of forcing here is that if the set of common neighbours of two vertices contains a $(k-1)$-clique then these two vertices are forced to have the same colour in any k-colouring.

Partition models
The *partition model* assumes that there is an (unknown) partition of the vertices into k blocks, edges between blocks are present independently with some probability p, and there are no edges within blocks. In this model we may also insist that the partition is 'balanced', in that each block size is $\Omega(n)$, or that the block sizes are nearly or exactly equal, or that the partition is generated at random. The partition model with p about $\frac{1}{2}$ and nearly equal block sizes is similar to the uniform k-colourable model – see [46]. Most results are phrased in terms of the partition model.

From now on we shall be interested in polynomial time algorithms that succeed with high probability. Given sufficiently low failure probabilities, it may be possible to use such algorithms as the first stage of a polynomial expected time algorithm as above – see Fürer, Subramanian and Veni Madhavan [80].

The first stage in the Dyer-Frieze algorithm above was successful with high probability since random k-colourable graphs are dense. It is more of a challenge to k-colour sparser k-colourable graphs, and several investigations have involved the partition model with $p = p(n) \to 0$ as quickly as possible (with $k = 3$ in particular). There is also interest in the dense case with $k = k(n)$ growing as fast as possible.

Partition models: sparse case
There are various methods based on ideas like counting degrees or numbers of common neighbours (as with the mingreedy approach). These methods will work with high probability for the partition model with not too small probability p and say $k = 3$, as long as we ensure that the blocks of the partition are nearly equal in size. For example, if block sizes are nearly equal and $p \geq n^{-\frac{1}{2}+\epsilon}$, then usually vertices in the same block will have more common neighbours than vertices in different blocks: if we extend this idea and count paths of length more than 2 then we can colour with high probability for $p \geq n^{-1+\delta}$ – see Blum and Spencer [19].

Petford and Welsh [165] report interesting simulation results for a Metropolis-type random recolouring method, for a range of probabilities p (see also Zerovnik [206, 207]).

A very different method allows the probability p to be pushed down to c/n for a (large) constant c (for k constant and blocks of nearly equal size). This 'spectral technique' of Alon and Kahale [7] uses the fact that an approximation to the unknown colour classes can be read off from the eigenvectors corresponding to the two smallest eigenvalues of the adjacency matrix of the graph (with some high degree vertices omitted). This is the first phase. The approximate colouring is then refined in two further phases. In the second phase, vertices are repeatedly recoloured with the least popular colour amongst their neighbours. The third phase involves uncolouring vertices with too few neighbours of some different colour. With high probability, the vertices remaining coloured have the original (unordered) colouring, and the subgraph induced by the uncoloured vertices has each component of size $O(\log n)$; and so we may complete the colouring by exhaustive search.

Research Problem 18 *The analysis of [7] is not valid unless c is sufficiently large. Can the complete range of c be covered by a polynomial time algorithm?*

The partition model with many blocks
Suppose now that we let p be fixed and consider large k. Assume that all block sizes are $\Omega(n/k)$. Then two vertices in a given block B will tend to have more common neighbours than a vertex in B and one in another block. This idea can be used to design an algorithm that will yield a k-colouring with high probability as long as $k = o((n/\log n)^{\frac{1}{2}})$ Kučera [129]. In Kučera [130] it is shown that the basic greedy colouring algorithm **whp** uses about $n/\log n$ colours when k is about $\sim n^{\epsilon}$ for some $\epsilon > 0$.

The semi-random model
An extension of the partition model is the *semi-random model* – see Blum [18] and Blum and Spencer [19]. Here there is an unknown partition as above, with k fixed; but now an adversary runs through the pairs of vertices from different blocks and decides for each whether or not it should be an edge. Once the adversary has made her choice for a particular possible edge then that choice is reversed with probability p (the noise rate). The hope is to find methods that work even for low noise rates.

In this model, counting methods such as those mentioned for the sparse partition model will not work, but methods based on the idea of forcing still do work. Turner's [198] no-choice algorithm works when $p \geq n^{-\frac{1}{k}+\epsilon}$ and each block size is $\Omega(n)$. The 'two-stage' algorithm of Blum and Spencer

[19] works for the same probabilities for the case $k = 3$ without restrictions on the block sizes. (If BPP $\not\supseteq$ NP then we cannot handle $k \geq 4$ without some restriction on the block sizes [19].)

An even more successful 'forcing' algorithm is the algorithm Link of [19], which works as follows. (A similar algorithm and results are presented independently in Fürer and Subramanian [79].) Given a graph G and integer $k \geq 3$, create a graph H with the same vertices, and with two vertices adjacent if in G the common neighbours contain a $(k-1)$-clique. If H has exactly k components then output these as the colour classes for G. For $k = 3$ the algorithm Link works with high probability for $p \geq n^{-\frac{3}{5}+\epsilon}$ and block sizes $\Omega(n)$. The noise rate p here is sufficient to ensure that the probability that two vertices from the same block are adjacent in H is well above the threshold for connectivity. A similar result holds for any constant k.

Recently, Subramanian [191] has found polynomial expected time algorithms for colouring in the semi-random model.

Research Problem 19 *Can the eigenvector approach of [7] be extended to the semi-random model?*

5.4 Other graph colouring problems

Frieze, Jackson, McDiarmid and Reed [67] considered the edge chromatic number (or chromatic index) of $G_{n,p}$ (with p fixed). They proved an upper bound of $n^{-\Omega(np)}$ for the probability that this equals the maximum degree plus 1. The result is obtained by analysing an algorithm that repeatedly rips out near perfect matchings. This was recently significantly strengthened by Perkovic and Reed [164] who describe a polynomial expected time algorithm for optimally edge colouring $G_{n,1/2}$. This is not straightforward, as the deterministic algorithm of last resort needed to colour graphs on which the first phase fails, runs in $2^{\Omega(n^2)}$ time.

Corresponding results for the total chromatic number are proved in McDiarmid and Reed [155] by analysing an algorithm that incorporates the edge colouring algorithm of [67].

5.5 Colouring random hypergraphs

A *weak colouring* of a hypergraph is a colouring of the points so that no edge is monochromatic, and in a *strong colouring* no two points in the same edge have the same colour. In Schmidt-Pruzan, Shamir and Upfal [180] it is shown that a natural hypergraph version of the modified greedy approach works well for weak colourings of suitable random hypergraphs. For strong colourings, a three stage method works well, Schmidt-Pruzan [179]. In the first stage we pick a random partition of the vertex set into sufficiently many blocks so that for each block we expect the edge-fragments of size at least three to form a sparse random hypergraph. Such sparse hypergraphs are easy to colour using a breadth-first search idea. We thus obtain a more refined partition. Finally, on each block of the refined partition we have a random graph and we can use the familiar greedy approach.

Chen and Frieze [36] applied the ideas of [7] to the problem of colouring a randomly generated (weakly) 2-colourable 3-uniform hypergraph and obtained similar results.

5.6 Enumerative approaches

Let x and y be non-adjacent vertices in a graph G. Let G'_{xy} denote the graph obtained by adding the edge joining x and y; and let G''_{xy} denote the graph obtained by 'contracting' x and y to form a single new vertex adjacent to each vertex which was adjacent to either x or y. Then by partitioning the colourings according to whether or not x and y have the same colour, we see that $\chi(G) = \min\{\chi(G'_{xy}), \chi(G''_{xy})\}$. This gives a 'branching' for a branch-and-bound approach to

determining $\chi(G)$. We may 'bound' by using $\chi(G) \geq \omega'(G)$, where $\omega'(G)$ is the size of some clique in G. Also, of course, we know the chromatic number of any complete graph.

For any graph G, let $Z^*(G)$ denote the minimum number of nodes in a corresponding search tree which determines $\chi(G)$. Consider the random graph $G_{n,p}$ with p constant. We might hope that usually $Z^*(G_{n,p})$ is not too big. However, we have the following result of McDiarmid [146], in the spirit of the result of Chvátal mentioned in Section 4. There is a constant $c > 0$ such that $Z^*(G_{n,p}) \geq \exp(cn(\log n)^{\frac{1}{2}})$ **whp**. It follows that any algorithm based on this branch-and-bound approach will be much slower asymptotically than methods based on enumerating stable sets — see McDiarmid [150].

There is a natural enumerative 'backtrack' colouring algorithm that may be seen to fall into the above framework. Wilf [204] showed however that if we simply try to test if $\chi(G_{n,p}) \leq 3$, then the method generates a search tree with expected number of nodes uniformly bounded. This is because with p constant it does not take long to run into a complete graph K_4 — see also Bender and Wilf [15], McDiarmid [150].

6 Graph Isomorphism

Suppose we are given two graphs $G_i = ([n], E_i)$, $i = 1, 2$. Can we quickly tell whether or not $G_1 \cong G_2$ i.e. is there a bijection $f : [n] \to [n]$ such that $vw \in E_1$ if and only if $f(v)f(w) \in E_2$? The exact complexity of this problem is not known and determining it remains an outstanding open problem. It is solvable in polynomial time for graphs of bounded degree, Luks [142]. The first result on the average case in this area is due to Babai, Erdős and Selkow [14]. They considered *canonical labellings* of the graph $G_{n,1/2}$. A canonical labelling is a partial function which maps a graph to an ordering of its vertices such that a bijection which preserves orderings is an isomorphism and vice-versa. In [14] they used the fact that **whp** the $r = \lceil 3 \log_2 n \rceil$ largest degrees are unique and that furthermore, the remaining vertices have distinct adjacency relationships with these r vertices.

This particular result was strengthened by Karp [109], Lipton [137] and Babai and Kučera [13]. In particular, Babai and Kučera describe an algorithm which runs in $O(n^2)$ expected time on $G_{n,1/2}$. The algorithm describes a way of canonically labelling all but an extremely small proportion of the graphs on vertex set $[n]$. At a given stage the algorithm will have produced an ordered partition $V_1, V_2, \ldots, V_h$ of $[n]$ which is then refined. Each set V_i of the partition is subdivided according to the number of neighbours a given vertex has in each of the other sets $V_j, j \neq i$. After only two such rounds we should find that $h \geq n - \log n / \log \log n$. There then follows a short enumerative search for the *lexicographically first* order consistent with what has been constructed so far. This suffices on the overwhelming majority of graphs.

Kučera [128] considered the case of $G_{n,r}$ where r is fixed. He describes an $O(m)$ (= *linear*) expected time algorithm, where $m = rn/2$ is the number of edges. The degree r is fixed here and Luks' result applies, but Kučera's algorithm is faster and simpler. It is based on the idea that if M is the set of vertices on shortest cycles of $G_{n,r}$ then no two vertices are likely to possess the same matrix $(d_{ij}(v))$ where $d_{ij}(v)$ is the number of vertices at distance i from v and j from M.

7 Network Flow and Related Problems

7.1 Network Flow

The network flow problem is one of the most studied problems in combinatorial optimization. The fastest known algorithm runs in $O(mn \log n)$ time - Alon [5] - on a network with n vertices and m

edges.

Some early work studied the complete graph K_n with iid random edge capacities, Karp [107], Grimmett and Welsh [90] and Grimmett and Suen [89]. In particular they show that **whp** the minimum cut is the set of edges incident with the source or sink and so the maximum flow will be asymptotically equal to $n\mu$, where the expected edge capacity μ is independent of n.

Now consider the case of a single source and sink and edge capacities equal to 0 or 1. The problem now reduces to that of finding a maximum number of edge disjoint paths from the source to the sink. So consider $G_{n,m}$ and the problem of finding the maximum number of edge disjoint paths between vertex 1 and vertex n.

Suen [192] – see also Grimmett [86], Grimmett and Kesten [87] – studied the case where $m \approx cn$ for constant $c > 0$. The minimum cut is no longer the set of edges incident with the source or sink. The parts of the network that are near the source and sink resemble two branching trees. The maximum flow is distributed as the minimum of two maximum flows through two branching trees (one each for the source and sink).

These results are existential rather than algorithmic. Assume next that m is above the threshold for connectivity. Hochbaum [95] gave an $O(n)$ time algorithm for this problem, which does not of course look at the whole graph. It is based on growing disjoint forests rooted at the sources and sinks until the corresponding trees are large enough so that a matching can be found, pairing up trees from different forests. Motwani [163] proved that Dinic's classical algorithm [45] runs **whp** in time $O(m \log n / \log d)$ due to the fact that **whp** every non-maximum flow has an augmenting path of length $O(\log n / \log d)$.

7.2 Transportation Problem

Hassin and Zemel [93] studied a probabilistic version of this problem where the underlying network is a complete bipartite graph, the supplies and demands are arbitrary and the edge capacities are random. They give conditions under which the problem is likely to have a feasible solution. Karp, Motwani and Nisan [115] discuss linear time algorithms for random transportation problems. Again it is the capacities which are random, the supplies and demands are required to satisfy certain conditions. The algorithms are based on the idea that a random instance is likely to be close to some fixed deterministic relaxation, which can be quickly solved and whose solution can be used to help find a solution to the random instance.

Kučera [133] considered the case of an *arbitrary* planar directed network in which the edge capacities are chosen uniformly and independently from $[0, b]$, for some $b > 0$. A careful implementation of an algorithm of Itai and Shiloach [96] is shown to run in *linear* expected time.

Karp [110] studied random assignment problems with independent weights. A nice modification of a standard shortest augmenting path algorithm, [51] or [197], is shown to run in $O(mn \log n)$ expected time.

7.3 Disjoint Path Problems

Suppose we are given a graph $G = (V, E)$ and a set of pairs (a_i, b_i), $1 \le i \le K$ of vertices. In the Edge Disjoint Paths Problem (EDPP) we want to find paths P_i joining *source* a_i to *sink* b_i for $1 \le i \le K$ which are edge disjoint, or prove it is not possible. In the Vertex Disjoint Paths Problem (VDPP), the vertices are all distinct and we want vertex disjoint paths. Both problems are solvable in polynomial time if K is fixed, independent of the input, Robertson and Seymour [176], but NP-hard if K varies. The problem is interesting for theoretical and practical reasons; the latter interest comes from its use as a model for some communications problems.

For random graphs $G_{n,m}$ the VDPP was considered by Shamir and Upfal [186] who gave a linear time algorithm which **whp** succeeds in finding paths provided $m \geq 2n \log n$ and $K = O(\sqrt{n})$. It should be remarked that here the two sets of vertices are fixed *before* the random graph is constructed. The problem was also considered by Hochbaum [95] who gave a $o(m)$ time algorithm when $K = O(\sqrt{d/\log n})$, $d = 2m/n$ is the average degree. Both algorithms are based on growing disjoint forests rooted at the sources and sinks until the corresponding trees are large enough so that for each i the tree rooted at a_i can be joined to the tree rooted at b_i.

The above approach is simple and efficient, but does not address the problem when the random graph is constructed first and then the sources and sinks are chosen by an *adversary*. Suppose $2m/n - \log n \to \infty$ so that $G_{n,m}$ is connected **whp**. Let D be the median distance between pairs of vertices in $G_{n,m}$. Then $D = O(\log n/\log d)$ **whp**. Clearly it is not possible to connect more than $O(m/D)$ pairs of vertices by edge-disjoint paths, for all choices of pairs, since some choice would require more edges than all the edges available. Also, some restriction on the number of times a vertex can be a source or sink is necessary. Thus the following theorem of Broder, Frieze, Suen and Upfal [30] is optimal up to constant factors.

Theorem 19 *Suppose $2m/n - \log n \to \infty$. Then there exist positive constants α and β such that* **whp**, *for all $A = \{a_1, a_2, \ldots, a_K\}, B = \{b_1, b_2, \ldots, b_K\} \subseteq [n]$ satisfying*

(i) $K = \lceil \alpha m \log d/\log n \rceil$,

(ii) for each vertex v, $|\{i : a_i = v\}| + |\{i : b_i = v\}| \leq \min\{d_G(v), \beta d\}$,

there exist edge-disjoint paths in $G_{n,m}$, joining a_i to b_i, for each $i = 1, 2, \ldots, K$. Furthermore, there is an $O(nm^2)$ time randomized algorithm for constructing these paths.

The strategy for proving Theorem 19 is quite different from [186] and [95]. First of all the sources and sinks are joined, by a network flow algorithm, to randomly chosen $\tilde{a}_i, \tilde{b}_i$, $1 \leq i \leq K$. This has a *spreading out* effect, similar to that achieved by the method of Valiant and Brebner [201] for routing messages in the n-cube. The new sources and sinks are then joined up by utilizing random walks.

Research Problem 20 *Can the results of Theorem 19 be extended to $G_{n,r}$ – in particular can one* **whp** *link $K \leq crn/\log r$ arbitrary pairs of vertices by edge disjoint paths for some constant $c > 0$? Here we assume r is constant as $n \to \infty$.*

The VDPP is discussed in [31]. Using similar ideas to those above it is shown that:

Theorem 20 *Suppose $2m/n - \log n \to \infty$. Then there exist positive constants α, β such that* **whp**, *for all $A = \{a_1, a_2, \ldots, a_K\}, B = \{b_1, b_2, \ldots, b_K\} \subseteq [n]$ satisfying*

(i) $A \cap B = \emptyset$,

(ii) $|A| = |B| = K \leq \frac{\alpha n \log d}{\log n}$,

(iii) $|N(v) \cap (A \cup B)| \leq \beta |N(v)|, \qquad \forall v \in V$,

there are vertex disjoint paths P_i from a_i to b_i for $1 \leq i \leq K$. Furthermore, there is an $O(nm^2)$ time randomized algorithm for constructing these paths.

Here $N(v)$ is the neighbour set of vertex v. This is again optimal up to the constant factors α, β.

8 Shortest Paths and Minimum Spanning Trees

8.1 Shortest Paths

Most work in this area has been restricted to that of finding shortest paths between all pairs of nodes in a complete digraph with independently chosen random non-negative edge weights. More generally, one considers distributions which are *endpoint independent*. Loosely, this means that if the edges leaving a vertex are sorted according to their cost, then the associated endpoints occur in random order. Spira [188] showed that using a heap in a version of Dijkstra's algorithm [44] gave a solution in $O(n^2(\log n)^2)$ expected time. This was improved by Bloniarz [16] and Frieze and Grimmett [66]. Moffatt and Takaoka [160] subsequently reduced the expected running time to $O(n^2 \log n)$. Recently, Mehlhorn and Priebe [157] show this algorithm runs in time $O(n^2 \log n)$ **whp** and not just in expectation. They also give an $\Omega(n \log n)$ lower bound for the single source problem under a class of distributions.

Luby and Ragde [139] consider the problem of finding a single shortest path between a source s and a sink t. They show that searching simultaneously from *both* s and t can be efficient on average. For example they give a $\Theta(\sqrt{n} \log n)$ time bound assuming sorted edge lists and edge lengths chosen independently from "reasonable" distributions.

The are only a few papers we know of that deal with arbitrary, as opposed to non-negative weights. Kolliopolous and Stein [124] modify the Bellman-Ford dynamic programming algorithm and show that a single source problem can be solved in $O(n^2 \log n)$ expected time when the distribution is endpoint independent. Their model allowed negative cycles. Cooper, Frieze, Mehlhorn and Priebe [41] consider a model in which the arc costs $c_{i,j}$ are generated from

$$c_{i,j} = -u_i + u_j + v_{i,j},$$

where $v_{i,j} \geq 0$. It is assumed that the $v_{i,j}$'s are independent, identically distributed, bounded and their common probability function F satisfies $F'(0) > 0$. The u_i's are independent and bounded with mean zero. The algorithm does not see the u's and v's, only the values $c_{i,j}$. They show that a single source shortest path problem can be solved in $O(n^2)$ expected time and an all pairs shortest path problem can be solved in $O(n^2 \log n)$ expected time.

Research Problem 21 *Determine to what extent the results on non-negative distributions can be extended to the case where negative weight edges are allowed, but negative cycles are unlikely.*

Research Problem 22 *Find a $o(n^2 \log n)$ expected time algorithm for the all pairs problem under a natural class of distributions e.g. i.i.d. uniform on [0,1].*

Various authors have considered the lengths of shortest paths when the edge weights are random. For example, building on work of Kulkarni [135], David and Prieditis [43] have shown that if the edges of the complete digraph on vertex set $[n]$ are given edge lengths which are independent negative exponentials with mean μ then the expected distance between two fixed vertices is

$$\frac{1}{(n-1)\mu} \sum_{k=1}^{n-1} \frac{1}{k}.$$

8.2 Minimum Spanning Trees

Karp and Tarjan [119] considered the problem of finding a minimum weight forest in $G_{n,m}$ when the edge weights are iid. They give an $O(m+n)$ expected time algorithm based on the algorithm

of Cheriton and Tarjan [37] – see also McDiarmid [151]. Only recently has this been emulated for a randomised algorithm on an arbitrary graph – Karger, Klein and Tarjan [104]. Frieze [60] showed that, if the edges of the complete graph K_n has iid edge weights uniform on (0,1), then as $n \to \infty$ the expected weight of the minimum spanning tree tends to $\zeta(3) = \sum_{i=1}^{\infty} 1/i^3 \approx 1.2$. This was done by analysing the lengths of edges added by the greedy algorithm for constructing a minimum spanning tree. For related results see Timofeev [195], Frieze and McDiarmid [73], Avram and Bertsimas [12], Aldous [2, 3] and Janson [97]. Note that the paper [2] give 'local' properties of the random minimum spanning tree, such as the expected proportion of vertices a given degree.

Research Problem 23 *Determine the likely shape of a minimum spanning tree. For example, is the diameter of order* $\log n$ **whp***?*

McDiarmid, Johnson and Stone [154] investigate the behaviour of Prim's tree growing method for finding a minimum spanning tree, when applied for example to the complete graph with iid edge weights.

8.3 Connectivity

The paper of Karp and Tarjan [119] also shows that depth first search can be used to find all connected components or 2-connected components in $O(n)$ expected time. Thus in the average case one need not look at all edges. Karp [112] used similar algorithms to find the strong components of a random digraph in $O(n)$ expected time.

9 Graph Bisection

Here we are given a graph $G = (V, E)$ with n vertices, n even, and the problem is to find a partition of V into two equal sized subsets V_1, V_2 such that the number of edges between V_1 and V_2 is minimised. The minimum such number of edges is called the bisection width of G. If we take $G_{n,m}$ as a model of a random input then we find that **whp** all relevant cuts have $\approx m/2$ edges, provided m is sufficiently large, namely $m/n \to \infty$. Finding the exact bisection width **whp** in this model is still an open problem.

Positive results can be obtained if we consider sampling uniformly from $\mathcal{G}(n, m, b)$, the set of graphs with vertex set $[n]$, m edges and bisection width b. Basically the idea is that if b is significantly smaller than $m/2$ then there will be a unique bisection of size b and it will be easy to find. Dyer and Frieze [46] considered the case where $m = \Omega(n^2)$ and show how to solve the problem in polynomial expected time if $b \le (1-\epsilon)m/2$ for $\epsilon > 0$ fixed. The algorithm is based on comparing vertex degrees, but requires some results on *jumbled graphs* – Thomason [194] – to show that the algorithm can check its success in polynomial time.

Boppana [27] improved these results by giving a polynomial time algorithm when m is as small as $\Omega(n \log n)$. For a graph G on $[n]$ and $\mathbf{d} = (d_1, d_2, \ldots, d_n)$ let

$$f_G(\mathbf{d}, x) = \sum_{(i.j)\in E} \frac{1 - x_i x_j}{2} - \frac{1}{4} \sum_{i=1}^{n} d_i (x_i^2 - 1).$$

For $S \subseteq [n]$ let $x = x(S)$ be defined by $x_i = 1, i \in S$ and $x_i = -1, i \notin S$. Then $f_G(\mathbf{d}, x(S))$ is the number of edges in the cut $S, \bar{S}$. So in general $\min_x f_G(\mathbf{d}, x)$ is a lower bound on bisection width. Boppana shows that if

$$0 \le b \le m/2 - 5\sqrt{mn \log n} \tag{7}$$

then **whp** if $\mathbf{d}^*$ yields $\max_{\mathbf{d}} \min_x f_G(\mathbf{d}, x)$ then $b = f_G(\mathbf{d}^*, x^*) = \min_x f_G(\mathbf{d}^*, x)$ and furthermore $x^* = x(S^*)$ where S^* is the optimum bisection.

Jerrum and Sorkin [100] considered a similar model and showed that simulated annealing (actually the Metropolis algorithm) was also likely to solve the problem. The requirements for b, m are more stringent than (7), but the result is nevertheless interesting since simulated annealing is often used in practice.

An earlier paper of Bui, Chaudhuri, Leighton and Sipser [32] considered a similar model of regular graphs and used flow techniques to find the bisection.

10 Counting Problems

Counting is a pervasive human activity and there has been a good deal of research on problems involving counting occurences of combinatorial substructures of given structures. Consider for example the beautiful determinantal formula for the number of spanning trees of a graph, see for example Lovász [138]. On the other hand, Valiant [200] showed that it is hard to compute the number of matchings in a graph and defined the class #P.

Research Problem 24 *Find a #P-hard problem which is exactly solvable* **whp** *on some model of a random graph. (It is appropriate to exclude here trivial problems where the answer is zero* **whp***. For example, counting 3-colourings of $G_{n,1/2}$ would be such a case.)*

Karp and Luby [114] introduced the notion of a *Fully Polynomial Randomized Approximation Scheme* (FPRAS). Jerrum and Sinclair's paper on the 0-1 permanent[2][99] includes a proof that there is an FPRAS for counting perfect matchings in almost every $G_{n,m}$. (Motwani [163] filled in a gap near the threshold for the existence of perfect matchings). Frieze and Jerrum [68] showed that an algorithm due to Karmarker, Karp, Lipton, Lovász and Luby [105] for estimating a 0-1 permanent is also almost always an FPRAS. Here one computes a quantity X whose expectation is the permanent and the issue then is how large is the variance of X. Rasmussen [175] gave a particularly simple algorithm for estimating the permanent. It uses ideas similar to those found in Hammersley [92] and Knuth [121]. Here he computes a different quantity $Y = Y(A)$ whose expectation is the permanent and he shows that the variance is small **whp** provided $mn^{-3/2} \to \infty$ as $n \to \infty$. Y is computed recursively by $Y(A) = d_1 Y(A_1)$ where d_1 is the number of 1's in row 1 of A and A_1 is obtained by choosing a random 1 in row 1 and then deleting row 1 plus the corresponding column.

Frieze, Jerrum, Molloy, Robinson and Wormald [69] give an FPRAS for counting Hamilton cycles in almost every $G_{n,r}$, r fixed. They do this by reducing approximate counting of Hamilton cycles to approximate counting of 2-factors.

Research Problem 25 *Extend the result on counting Hamilton cycles in $G_{n,r}$ to the case where $r \to \infty$.*

Frieze and Suen [77] give an algorithm for approximately counting the number of Hamilton cycles in the random directed graph $D_{n,m}$ provided $mn^{-3/2} \to \infty$ as $n \to \infty$. Here approximate counting of Hamilton cycles is reduced to approximate counting of perfect matchings. Rasmussen [175] extended his scheme to cover counting directed Hamilton cycles in this range.

Research Problem 26 *Can the requirement $mn^{-3/2} \to \infty$ as $n \to \infty$ in [77] for counting Hamilton cycles in $D_{n,m}$ be substantially weakened?*

[2]The permanent of an $n \times n$ 0-1 matrix can be interpreted as the number of perfect matchings in an associated bipartite graph.

Alon, Frieze and Welsh [6] show that the Tutte polynomial $T_G(x,y)$ of the random graph $G_{n,p}$ can be evaluated in polynomial time **whp** for $x,y \geq 1$ and $p \geq K \log n/n$, $K = K(x,y)$.

Research Problem 27 *Improve the results of [6] on random graphs by removing the lower bound on p and extend the analysis to $G_{n,r}$.*

Research Problem 28 *Find an FPRAS for estimating any of the following in a random graph:*

(a) *the number of k-cliques.*

(b) *the number of trees of all sizes.*

(c) *the number of spanning forests.*

(d) *the number of acyclic orientations.*

(e) *the number of Euler tours.*

(f) *the number of cycles.*

The same caveat about trivial problems applies here as in Problem 24.

11 Parallel Algorithms

In this section we consider parallel algorithms that are efficient on average. Generally speaking we restrict our attention to cases where the worst-case parallel complexity seems bad, but the average case is good. The model of computation is usually the PRAM.

11.1 Hamilton Cycles and Matchings

Frieze [61] considered the problem of finding a Hamilton cycle in $G_{n,p}$ where $0 < p < 1$ is constant. A simple recursive patching algorithm was described which determines whether or not $G_{n,p}$ is Hamiltonian in $O((\log\log n)^2)$ expected time using $n(\log n)^2$ processors. Using more sophisticated computational techniques McKenzie and Stout [156] were able to reduce the expected running time to $O(\log^* n)$ and the number of processors to $n/\log^* n$, thus yielding an algorithm with optimal expected work.

Coppersmith, Raghavan and Tompa [42] considered the sparser case where the edge probability $p \geq c(\log n)/n$, for c sufficiently large. They give an algorithm which **whp** finds a Hamilton cycle in $O((\log n)^2)$ time and uses $n \log n$ processors. Their idea is to generate a (near) random 2-factor, which **whp** has $O(\log n)$ cycles and to then patch these cycles together by extension and rotation.

Shamir and Upfal [184] considered finding perfect matchings in a distributive model of computing. They show that if $m \geq Kn \log n$ for large enough K then **whp** a perfect matching can be found in $G_{n,m}$ in $O((\log n)^2)$ time.

The algorithm of Karmarker et al [105] for estimating the permanent involves computing a sequence of randomly chosen determinants. The analysis of Frieze and Jerrum [68] shows that this algorithm is implementable as an NC algorithm **whp** when the input matrix is $n \times n$ 0-1 with m randomly chosen 1's and $mn^{-3/2} \to \infty$ as $n \to \infty$.

Research Problem 29 *Can perfect matchings or Hamilton cycles be found in $O(\log n)$ time at the respective hitting times for their existence?*

11.2 Two lexicographic problems

11.2.1 Maximal independent sets

There are a number of efficient parallel algorithms for finding a maximal independent set in a graph. On the other hand the problem of finding the *lexicographically first* such set (LFMIS) is complete for P. Thus it is unlikely that an NC algorithm exists for this problem.

Coppersmith, Raghavan and Tompa [42] described a natural parallel version of the greedy maximal set algorithm, PARGREEDY, and showed that it runs in $O((\log n)^2/\log\log n)$ expected time on $G_{n,p}$, uniformly over p. Subsequently, Calkin and Frieze [33] showed that it actually runs in $O(\log n)$ time (uniformly over p). See also Calkin, Frieze and Kučera [34].

Recently Chen and Frieze [35] extended the result of [42] to hypergraphs. The model is $H(n,k,p)$ where each of the $\binom{n}{k}$ k-edges is chosen independently with probability p. They describe a parallel algorithm which runs in $O((\log n)^3/\log\log n)$ expected time (uniformly over p). The algorithm uses $O(m)$ processors, where $m \approx \binom{n}{k}p$ is the number of edges.

Research Problem 30 *Combine the analysis of [33] and [35] to reduce the upper bound on expected running time of the algorithm in [35].*

11.2.2 Depth First Search Trees

Similarly, the problem of finding a lexicographically first depth first search tree (LFDFST) of a graph G is complete for P. Dyer and Frieze [48] discuss finding the LFDFST of $G_{n,p}$, $0 < p < 1$ constant. Here **whp** the LFDFST consists of the lexicographically first maximal path (LFMP), which is of length $n - o(\log n)$ plus a few vertices attached near one end. **Whp** the LFMP splits into small pieces which can be found in parallel and put together. Their algorithm uses n processors and runs in $O((\log n)^2)$ expected time.

Research Problem 31 *Extend the analysis to the case $p \to 0$.*

11.3 Vertex Colouring

As already mentioned in Section 5, in our current state of knowledge, we can only efficiently colour random graphs with about twice the number of colours that are really needed i.e. find 2-optimal colourings.

An early paper of Shamir and Upfal [185] considered colouring random graphs in a distributed model of computation. Their algorithm runs in $O(np\log n)$ rounds on $G_{n,p}$ and gives a $(2+\delta)$-optimal colouring. Frieze and Kučera [71] considered $G_{n,p}$ where $0 < p < 1$ is constant. Using a parallel version of the standard greedy algorithm they show that such a graph can **whp** be $(2+\delta)$-optimally coloured in $O((\log n)^{2+o(1)})$ time using $O(n^2/(\log n)^2)$ processors. Coppersmith, Raghavan and Tompa [42] gave a more complex algorithm which works **whp** for $p \to 0$ as well. It runs in $o((\log n)^5)$ time and uses $O(n^2p)$ processors.

Research Problem 32 *Find a parallel algorithm which $(2+\delta)$-optimally colours $G_{n,p}$ but improves the running time of the algorithm of [42].*

Kučera [132] considered the random k-colourable model for which efficient sequential optimal colouring algorithms are known to work **whp**. He showed that if $k = (\log n)^{O(1)}$ and the graph G is drawn uniformly from the set of k-colourable graphs with n vertices and $\Omega(n^2p)$ random edges (p constant) then his algorithm will k-colour G in *constant* expected time.

11.4 Shortest paths

Frieze and Rudolph [76] considered the all-pairs shortest paths problem when the edge lengths are independent uniform [0,1]. They showed that a standard matrix product algorithm can be implemented in $O(\log\log n)$ time using n^3 processors.

References

[1] M. Ajtai, J. Komlós and E. Szemerédi. *The first occurrence of Hamilton cycles in random graphs.* Annals of Discrete Mathematics 27 (1985) 173-178.

[2] D. Aldous, *A random tree model associated with random graphs*, Random Structures and Algorithms 1 (1990), 383-402.

[3] D. Aldous, *Asymptotic fringe distributions for general families of random trees*, Annals of Applied Probability 1 (1991) 228-266.

[4] D.Aldous, *Asymptotics in the Random Assignment Problem*, Probability Theory and Related Fields 93 (1992) 507-534.

[5] N. Alon, *Generating pseudo-random permutations and maximum flow algorithms*, Information Processing Letters 35 (1990) 201-204.

[6] N.Alon, A.M.Frieze and D.Welsh, *Polynomial time randomised approximation schemes for Tutte-Gröthendieck invariants: the dense case*, Random Structures and Algorithms 6 (1995) 459-478.

[7] N. Alon and N. Kahale, *A spectral technique for coloring random 3-colorable graphs*, Proceedings of the 26th Annual ACM Symposium on the Theory of Computing (1994) 346-355.

[8] N. Alon and J.H. Spencer, *The Probabilistic Method,* Wiley, 1992.

[9] D. Angluin and L.G. Valiant, *Fast probabilistic algorithms for Hamilton circuits and matchings*, Journal of Computer and System Sciences 18 (1979) 155-193.

[10] J. Aronson, A.M. Frieze and B. Pittel, *Finding matchings in sparse random graphs*, in preparation.

[11] D. Avis and C.W. Lai, *The Probabilistic Analysis of a Heuristic for the Assignment Problem*, SIAM Journal on Computing 17 (1988) 732-741.

[12] F. Avram and D. Bertsimas, *The minimum spanning tree constant in geometric probability and under the independent model: a unified approach*, Annals of Applied Probability 2 (1992) 113-130.

[13] L.Babai and L.Kucera, *Canonical labelling of graphs in linear average time*, Proceedings of the 20th Annual IEEE Symposium on the Foundations of Computer Science (1979) 39-46.

[14] L.Babai, P.Erdős and S.M.Selkow, *Random graph isomorphisms*, SIAM Journal on Computing 9 (1980) 628-635.

[15] E.A. Bender and H.S. Wilf, *A theoretical analysis of backtracking in the graph coloring problem,* J. Algorithms 6 (1985) 275-282.

[16] P. Bloniarz, *A shortest-path algorithm with expected time* $O(n^2 \log n \log^* n)$*,* SIAM Journal on Computing 12 (1983) 588-600.

[17] P.A. Bloniarz, M.J. Fisher and A.R. Meyer, *A note on the average time to compute transitive closures,* Automata, Languages and Programming, S. Michaelson and R. Milner (eds.), Edinburgh University Press, Edinburgh, 1976, 425-434.

[18] A. Blum, *Some tools for approximate 3-coloring,* Proceedings of the 31st IEEE Symposium FOCS (1990) 554-562.

[19] A.Blum and J.Spencer, *Coloring random and semi-random k-colorable graphs*, Journal of Algorithms 19 (1995) 204-234.

[20] B.Bollobás, *Almost all regular graphs are Hamiltonian*, European Journal on Combinatorics 4 (1983) 97-106.

[21] B.Bollobás, *Random Graphs,* Academic Press, 1985.

[22] B. Bollobás, *The chromatic number of random graphs,* Combinatorica 8 (1988) 49-55.

[23] B. Bollobás. *The evolution of sparse graphs.* Graph Theory and Combinatorics. (Proc. Cambridge Combinatorics Conference in Honour of Paul Erdős (B. Bollobás; Ed)) Academic Press (1984) 35-57.

[24] B.Bollobás and G.Brightwell, *The width of random graph orders*, The Mathematical Scientist 20 (1995) 69-90.

[25] B.Bollobás and P.Erdős, *Cliques in random graphs*, Mathematical Proceedings of the Cambridge Philosophical Society 80 (1976) 419-427.

[26] B.Bollobás, T.I.Fenner and A.M.Frieze, *An algorithm for finding Hamilton paths and cycles in random graphs*, Combinatorica 7 (1987) 327-341.

[27] R.Boppana, *Eigenvalues and graph bisection: an average case analysis*, Proceedings of the 28th Annual IEEE Symposium on the Foundations of Computer Science (1987) 327-342.

[28] A.Z.Broder, A.M.Frieze and E.Shamir, *On hidden Hamilton cycles*, Proceedings of the 23rd Annual ACM Symposium on Theory of Computing (1991) 182-189.

[29] A.Z.Broder, A.M.Frieze and E.Upfal, *On the satisfiability and maximum satisfiability of random 3-CNF formulas*, Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms (1993) 322-330.

[30] A.Z.Broder, A.M.Frieze, S.Suen and E.Upfal, *Optimal construction of edge disjoint paths in random graphs*, Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (1994) 603-612.

[31] A.Z.Broder, A.M.Frieze, S.Suen and E.Upfal, *Optimal construction of vertex disjoint paths in random graphs*, to appear in the proceedings of SODA '96.

[32] T.Bui, S.Chaudhuri, T.Leighton and M.Sipser, *Graph bisection with good average case behaviour*, Combinatorica 7 (1987) 171-192.

[33] N.Calkin and A.M.Frieze, *Probabilistic analysis of a parallel algorithm for finding maximal independent sets*, Random Structures and Algorithms 1 (1990) 39-50.

[34] N.Calkin, A.M.Frieze and L.Kučera, *On the expected performance of a parallel algorithm for finding maximal independent sets*, Random Structures and Algorithms 3 (1992) 215-222.

[35] H.Chen and A.M.Frieze, *Analysis of Parallel Algorithms for Finding A Maximal Independent Set in A Random Hypergraph*, submitted for publication.

[36] H.Chen and A.M.Frieze, *Coloring Bipartite Hypergraphs*, submitted for publication.

[37] D.Cheriton and R.E.Tarjan, *Finding minimum spanning trees*, SIAM Journal of Computing 5 (1976) 724-742.

[38] V.Chvátal, *Determining the stability number of a graph*, SIAM Journal on Computing 6 (1977) 643-662.

[39] V. Chvátal, *Almost all graphs with 1.44n edges are 3-colourable,* Random Structures and Algorithms 2 (1991) 11-28.

[40] C.Cooper and A.M.Frieze, *Hamilton cycles in random graphs and directed graphs*, submitted to Random Structures and Algorithms.

[41] C.Cooper, A.M.Frieze, K.Mehlhorn and V.Priebe, *On the shortest path problem in digraphs with random costs*, to appear.

[42] D.Coppersmith, P. Raghavan and M.Tompa, *Parallel graph algorithms that are efficient on average*, Proceedings of the 28th Annual IEEE Symposium on the Foundations of Computer Science (1987) 260-270.

[43] R.David and A. Prieditis, *The expected length of a shortest path*, Information Processing Letters 46 (1993) 135-141.

[44] E.Dijkstra, *A note on two problems in connection with graphs*, Numerische Mathematische 1 (1959) 269-271.

[45] E.A.Dinic, *Algorithm for solution of maximum flow in networks with power estimation*, Soviet Mathematics Doklaidy 11 (1970) 1277-1280.

[46] M.E.Dyer and A.M.Frieze, *Fast algorithms for some random NP-hard problems*, Journal of Algorithms 10 (1989) 451-489.

[47] M.E.Dyer and A.M.Frieze, *On patching algorithms for random asymmetric travelling salesman problems*, Mathematical Programming 46 (1990) 361-378.

[48] M.E.Dyer and A.M.Frieze, *A parallel algorithm for finding the lexicographically first depth first search tree in a random graph*, Random Structures and Algorithms 2 (1991) 233-240.

[49] M.E.Dyer, A.M.Frieze and C.J.H.McDiarmid, *Linear programs with random costs*, Mathematical Programming 35 (1986) 3-16.

[50] M.E.Dyer, A.M.Frieze and B.Pittel, *On the average performance of the greedy algorithm for finding a matching in a graph*, Annals of Applied Probability 3 (1993) 526-552.

[51] J.Edmonds and R.M.Karp, *Theoretical improvements in algorithmic efficiency for network flow problems*, Journal of the Association for Computing Machinery 19 (1972) 248-264.

[52] P. Erdős and A. Rényi, *On the evolution of random graphs*, Publ. Math. Inst. Hungar. Acad. Sci. 5 (1960) 17-61.

[53] P. Erdős and A. Rényi, *On the existence of a factor of degree one of a connected random graph*, Acta. Math. Acad. Sci. Hungar. 17 (1966) 359-368.

[54] P. Erdős and A. Rényi, *On random matrices*, Publ. Math. Inst. Hungar. Acad. Sci. 8 (1964) 455-461.

[55] P. Erdős and J.H. Spencer, *Probabilistic Methods in Combinatorics,* Academic Press, 1974.

[56] W. Fernandez de la Vega, *On the chromatic number of sparse random graphs,* Graph Theory and Combinatorics, ed B. Bollobás, Academic Press (1984) 321-328.

[57] W. Fernandez de la Vega, *Random graphs almost optimally colourable in polynomial time,* Annals of Discrete Mathematics 28 (1985) 311-317.

[58] T.I.Fenner and A.M.Frieze, *Hamiltonian cycles in random regular graphs*, Journal of Combinatorial Theory B 37 (1984) 103-112.

[59] A.M. Frieze, *Maximum matchings in a class of random graphs*, Journal of Combinatorial Theory Series B 40 (1986) 196-212.

[60] A.M. Frieze, *On the value of a random minimum spanning tree problem,* Discrete Applied Mathematics 10 (1985) 47-56.

[61] A.M. Frieze, *Parallel algorithms for finding Hamilton cycles in random graphs*, Information Processing Letters 25 (1987) 111-117.

[62] A.M. Frieze, *On the exact solution of random travelling salesman problems with medium-sized integer costs*, SIAM Journal on Computing 16 (1987) 1052-1072.

[63] A.M. Frieze, *Finding Hamilton cycles in sparse random graphs*, Journal of Combinatorial Theory B 44 (1988) 230-250.

[64] A.M. Frieze, *An algorithm for finding Hamilton cycles in random digraphs*, Journal of Algorithms 9 (1988) 181-204.

[65] A.M. Frieze, *On the independence number of random graphs,* Discrete Mathematics 81 (1990) 171-175.

[66] A.M. Frieze and G.R. Grimmett, *The shortest path problem for graphs with random arc-lengths*, Discrete Applied Mathematics 10 (1985) 57-77.

[67] A.M. Frieze, B. Jackson, C. McDiarmid and B. Reed, *Edge-colouring random graphs*, Journal of Combinatorial Theory B 45 (1988) 135-149.

[68] A.M. Frieze and M.R. Jerrum, *An analysis of a Monte Carlo algorithm for estimating the permanent*, Combinatorica 15 (1995) 67-83.

[69] A.M. Frieze, M.R. Jerrum, M. Molloy, R.W. Robinson and N.C. Wormald, *Generating and counting Hamilton cycles in random regular graphs*, to appear.

[70] A.M. Frieze, R.M. Karp and B. Reed, *When is the assignment bound asymptotically tight for the asymmetric traveling-salesman problem?*, SIAM Journal on Computing 24 (1995) 484-493.

[71] A.M. Frieze and L. Kučera, *Parallel colouring of random graphs*, in Proceedings of Random Graphs '87, Edited by M.Karonski, J.Jaworski and A.Rucinski, John Wiley and Sons (1990) 41-52.

[72] A.M. Frieze and T,. Łuczak, *On the independence and chromatic numbers of random regular graphs,* Journal of Combinatorial Theory B 54 (1992) 123-132.

[73] A.M. Frieze and C. McDiarmid, *On random minimum length spanning trees,* Combinatorica 9 (1989) 363-374.

[74] A.M. Frieze and B.Pittel, *Probabilistic analysis of an algorithm in the theory of markets in indivisible goods*, Annals of Applied Probability 5 (1995) 768-808.

[75] A.M. Frieze, A.J. Radcliffe and S. Suen, *Analysis of a simple greedy matching algorithm on random cubic graphs*, Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms (1993) 341-351.

[76] A.M. Frieze and L. Rudolph, *A parallel algorithm for all-pairs shortest paths in a random graph*, Proceedings of the Twenty-Second Allerton Conference on Communication, Control and Computing (1985) 663-670.

[77] A.M. Frieze and S. Suen, *Counting Hamilton cycles in random directed graphs*, Random Structures and Algorithms 3 (1992) 235-242.

[78] A.M. Frieze and S. Suen, *On the independence number of random cubic graphs*, Random Structures and Algorithms 5 (1994) 649-664.

[79] M. Fürer and C.R. Subramanian, *Coloring random graphs,* Proceedings of the Third Scandinavian Workshop on Algorithm Theory – SWAT '92, Springer-Verlag LNCS 621 (1992) 284-291.

[80] M. Fürer, C.R. Subramanian and C.E. Veni Madhavan, *Coloring algorithms for random graphs with polynomial expected running time,* 1994.

[81] D.Gale and S.Shapley, *College admissions and the stability of marriage*, American Mathematical Monthly 69 (1962) 9-15.

[82] P.G. Gazmuri, *Independent sets in random sparse graphs,* Networks 14 (1984) 367-377.

[83] M.X.Goemans and M.S.Kodialam, *A lower bound on the expected cost of an optimal assignment*, Mathematics of Operations Research 18 (1993) 267-274.

[84] O. Goldschmidt and D.S. Hochbaum, *A fast perfect-matching algorithm in random graphs*, SIAM Journal on Discrete Mathematics 3 (1990) 48-57.

[85] O. Goldschmidt and D.S. Hochbaum, *Asymptotically optimal linear algorithm for the minimum k-cut in a random graph*, SIAM Journal on Discrete mathematics 3 (1990) 58-73.

[86] G.R. Grimmett, *Random flows: network flows and electrical flows through random media*, in Surveys in Combinatorics 1985, ed. I. Anderson, Cambridge University Press, 1985, 59-96.

[87] G.R. Grimmett and H. Kesten, *First passage percolation, network flows and electrical resistances*, ZWT 66 (1984) 335-366.

[88] G.R. Grimmett and C.J.H. McDiarmid, *On colouring random graphs*, Proceedings of the Cambridge Philosophical Society 77 (1975) 313-324.

[89] G.R. Grimmett and S. Suen, *The maximal flow through a directed graph with random capacities*, Stochastics 8 (1982) 153-159.

[90] G.R. Grimmett and D.J.A. Welsh, *Flows in networks with random capacities*, Stochastics 7 (1982) 205-229.

[91] Y Gurevich and S. Shelah, *Expected computation time for Hamiltonian path problem*, SIAM Journal on Computing 16 (1987) 486-502.

[92] J.M.Hammersley, *Existence theorems and Monte Carlo methods for the momomer-dimer problem*, Research Papers in Statistics (1977) 125-146.

[93] R. Hassin and E. Zemel, *Probabilistic analysis of the capacitated transportation problem*, Mathematics of Operations Research 13 (1988) 80-89.

[94] M. Held and R.M. Karp, *A dynamic programming approach to sequencing problems*, SIAM Journal on Applied Mathematics 10 (1962) 196-210.

[95] D.S. Hochbaum, *An exact sub-linear algorithm for the max-flow, vertex-disjoint paths and communication problems on random graphs*, Operations Research 40 (1992) 923-935.

[96] A. Itai and Y. Shiloach, *Maximum flow in planar networks*, SIAM Journal on Computing 8 (1979) 135-150.

[97] S. Janson, *The minimal spanning tree in a complete gra[ph and a functional limit theorem for trees in a random graph*, Random Structures and Algorithms (1995) 337-355.

[98] M.R. Jerrum, *Large cliques elude the Metropolis process,* Random Structures and Algorithms 3 (1992) 347-359.

[99] M.R.Jerrum and A.J.Sinclair, *Approximating the permanent*, SIAM Journal on Computing 18 (1989) 1149-1178.

[100] M.R.Jerrum and G.Sorkin, *Simulated annealing for graph bisection*, Proceedings of the 34th IEEE Symposium on Foundations of Computer Science (1993) 94-103.

[101] D.S. Johnson, *The NP-completeness column: an ongoing guide: eleventh edition,* Journal of Algorithms 5 (1984) 284-299.

[102] A. Johri and D.W. Matula, *Probabilistic bounds and heuristic algorithms for coloring large random graphs,* Technical Report 82-CSE-6, Southern Methodist University, Dallas, Texas, 1982.

[103] D.R. Karger, R. Motwani and M. Sudan, Approximate graph coloring by semidefinite programming, Proceedings of the 35th Annual IEEE Symposium on the Foundations of Computer Science (1994) 2-13.

[104] D.R.Karger, P.Klein and R.E.Tarjan, *A randomized linear time algorithm for finding minimum spanning trees*, Journal of the Association for Computing Machinery 42 (1995)321-328.

[105] N.Karmarker, R.M.Karp, R.Lipton, L.Lovász and M.Luby, *A Monte-Carlo Algorithm for Estimating the Permanent.* SIAM Journal on Computing 22 (1993) 284-293.

[106] R.M. Karp, *The probabilistic analysis of some combinatorial search algorithms,* in Algorithms and Complexity: New Directions and Recent Results, J.F. Traub, ed., Academic Press (1976) 1-19.

[107] R.M. Karp, *The probabilistic analysis of combinatorial optimization algorithms*, Proceedings of the 10th International Symposium on Mathematical Programming.

[108] R.M. Karp, *A patching algorithm for the non-symmetric traveling salesman problem*, SIAM Journal on Computing 8 (1979) 561-573.

[109] R.M. Karp, *Probabilistic analysis of a canonical numbering algorithm for graphs*, Proceedings of Symposia in Pure Mathematics, Volume 34, American Mathematical Society (1979) 365-378.

[110] R.M. Karp, *An algorithm to solve the $m \times n$ assignment problem in expected time $O(mn \log n)$*, Networks 10 (1980) 143-152.

[111] R.M. Karp, *An upper bound on the expected cost of an optimal assignment,* Discrete Algorithms and Complexity: Proceedings of the Japan-US Joint Seminar (D. Johnson et al., eds.) 1-4, Academic Press, New York, 1987.

[112] R.M. Karp, *The transitive closure of a random digraph*, Random Structures and Algorithms 1 (1990) 73-94.

[113] R.M. Karp, J.K. Lenstra, C. McDiarmid and A.H.G. Rinnooy Kan, *Probabilistic analysis of combinatorial algorithms: an annotated bibliography*, in Combinatorial Optimisation: Annotated Bibliographies, edited by M.O'hEigeartaigh, J K.Lenstra and A.H.G.Rinnooy Kan, Wiley, Chichester, 1985.

[114] R.M. Karp and M.Luby, *Monte-Carlo algorithms for enumeration and reliability problems*, Proceedings of the 24th IEEE Symposium on Foundations of Computer Science (1983) 56-64.

[115] R.M. Karp, R. Motwani and N. Nisan, *Probabilistic analysis of network flow algorithms*, Mathematics of Operations Research 18 (1993) 71-97.

[116] R.M. Karp, A.H.G. Rinnooy Kan and R.V. Vohra, *Average case analysis of a heuristic for the assignment problem*, Mathematics of Operations Research 19 (1994) 513-522.

[117] R.M. Karp and M. Sipser, *Maximum matchings in sparse random graphs*, Proceedings of the 22nd Annual IEEE Symposium on the Foundations of Computer Science (1981) 364-375.

[118] R.M. Karp and J.M. Steele, *Probabilistic analysis of heuristics,* in The traveling salesman problem: a guided tour of combinatorial optimization, E.L.Lawler, J.K.Lenstra, A.H.G.Rinnooy Kan and D.B.Shmoys Eds., Wiley, 1985.

[119] R.M. Karp and R.E. Tarjan, *Linear expected time algorithms for connectivity problems*, Journal of Algorithms 1 (1980) 374-393.

[120] S. Khanna, N. Linial and S. Safra, *On the hardness of approximating the chromatic number.*

[121] D.Knuth, *Estimating the efficiency of backtrack programs*, Mathematics of Computations 29 (1975) 121-136.

[122] D.Knuth, R.Motwani and B.Pittel, *Stable husbands*, Random Structures and Algorithms 1, (1990) 1-14.

[123] Ph.G. Kolaitis, H.J. Prömel and B.L. Rothschild, *K_{l+1}-free graphs: asymptotic structure and a 0-1 law,* Trans. American Math. Soc. 303 (1987) 637-671.

[124] S.G.Kolliopoulos and C.Stein, *Finding real-valued single-source shortest paths in $o(n^3)$ expected time*, Proceedings of the 5th Conference on Integer Programming and Combinatorial Optimization (1996) 94-104.

[125] J. Komlós and E. Szemerédi, *Limit distributions for the existence of Hamilton circuits in a random graph*, Discrete Mathematics 43 (1983) 55-63.

[126] J. Komlós and E. Szemerédi, *Hamiltonian cycles in random graphs*, Infinite and Finite Sets (A.Hajnal, R.Rado and V.T.Sós, Eds.) Colloq. Math. Soc. J.Bolyai, Volume 10, North Holland, Amsterdam (1975) 1003-1011.

[127] L. Kučera, *Expected behaviour of graph colouring algorithms,* Lecture Notes in Computer Science 56, Springer-Verlag (1977) 447-451.

[128] L.Kučera, *Canonical labelling of regular graphs in linear expected time*, Proceedings of the 28th Annual IEEE Symposium on the Foundations of Computer Science (1987) 271-279.

[129] L. Kučera, *Graphs with small chromatic numbers are easy to color*, Information Processing Letters 30 (1989) 233-236.

[130] L. Kučera, *The greedy coloring is a bad probabilistic algorithm,* J. Algorithms 12 (1991) 674-684.

[131] L. Kučera, *Expected complexity of graph partitioning problems*, Discrete Applied Mathematics 57 (1995) 193-212.

[132] L. Kučera, *Parallel colouring graphs with small chromatic number*, to appear in Random Structures and Algorithms .

[133] L. Kučera, *Finding a maximum flow in (s,t)-planar network in linear expected time*,

[134] L. Kučera and D.W. Matula, *An expose and merge algorithm and the chromatic number of a random graph,* in Random Graphs '87, J. Jaworski, M. Karonski and A. Rucinski, Eds., Wiley, New York (1990) 175-188.

[135] V. G. Kulkarni, *Shortest paths in networks with exponentially distributed arc lengths*, Networks 16 (1986) 255-274.

[136] P. Lehert, *Clustering by connected components in $O(n)$ expected time,* RAIRO Inform. 15 (1981) 207-218.

[137] R.J. Lipton, *The beacon set approach to graph isomorphism*

[138] L.Lovász, Combinatorial problems and exercises, Second edition, North-Holland, 1993.

[139] M. Luby and P. Ragde, *Bidirectional search is $O(\sqrt{n})$ faster then Dijkstra's shortest path algorithm*, Algorithmica 4 (1989) 551-567.

[140] T. Łuczak, *The chromatic number of random graphs,* Combinatorica **11** (1991) 45-54.

[141] T. Łuczak and J. Wierman, *The chromatic number of random graphs at the double-jump threshold,* Combinatorica 9 (1987) 275-284.

[142] E.M.Luks, *Isomorphism of bounded valence can be tested in polynomial time*, Proceedings of the 21st Annual IEEE Symposium on the Foundations of Computer Science (1980) 42-49.

[143] C. Lund and M. Yannakakis, *On the hardness of approximating minimization problems,* Proc 25th STOC (1993) 286-293.

[144] D.W. Matula, The largest clique size in a random graph, Technical Report, Department of Computer Science, Southern Methodist University, Dallas, Texas, 1976.

[145] D.W. Matula, *Expose-and-merge exploration and the chromatic number of a random graph,* Combinatorica **9** (1987) 275-284.

[146] C. McDiarmid, *Determining the chromatic number of a graph*, SIAM Journal on Computing 8 (1979), 1-14.

[147] C. McDiarmid, *Colouring random graphs badly,* Graph theory and Combinatorics, ed. R.J. Wilson, Pitman Research Notes in Mathematics **34**, Pitman, 1979, 76-86.

[148] C. McDiarmid, *Clutter percolation and random graphs*, Combinatorial Optimization, (V.J.Rayward-Smith, Ed.) Mathematical Programming Study 13 (1980) 17-25.

[149] C. McDiarmid, *Achromatic numbers of random graphs,* Mathematical Proceedings of the Cambridge Philosophical Society 92 (1982) 21-38.

[150] C. McDiarmid, *Colouring random graphs,* Annals of Operations Research **1** (1984) 183-200.

[151] C. McDiarmid, *On some conditioning results in the probabilistic analysis of algorithms*, Discrete Applied Mathematics 10 (1985), 197-201.

[152] C. McDiarmid, *Colourings of random graphs,* in R.Nelson and R.J.Wilson, eds., Graph Colouring, Pitman Research Notes in Mathematics **218**, Pitman, Longman Scientific and Technical, Essex, UK, 79-86, 1990.

[153] C. McDiarmid, *On the chromatic number of random graphs,* Random Structures and Algorithms 1 (1990) 435-442.

[154] C. McDiarmid, E.Johnson and H.Stone, *On finding a minimum spanning tree in a network with random weights*, to appear.

[155] C. McDiarmid and B. Reed, *On total colourings of graphs*, Journal of Combinatorial Theory B 57 (1993), 122-130.

[156] P.D. Mckenzie and Q.F. Stout, *Optimal parallel construction of Hamiltonian cycles and spanning trees in random graphs*,

[157] K.Mehlhorn and V.Prieber, *On the all pairs shortest path algorithm of Moffat and Takaoka*, preprint.

[158] S. Micali and V.V. Vazirani, *An $O(|V|^{1/2}|E|)$ time algorithm for finding maximum matchings in general graphs*, Proceedings of the 21st Annual ACM Symposium on Theory of Computing (1980) 17-27.

[159] D.L. Miller and J.F. Pekny, *Exact solution of large asymmetric traveling salesman problems*, Science 251 (1991) 754-762.

[160] A. Moffat and T. Takaoka, *An all pairs shortest path algorithm with expected time $O(n^2 \log n)$*, Proceedings of the 26th Annual IEEE Symposium on the Foundations of Computer Science (1985) 101-105.

[161] M. Molloy, *The chromatic number of sparse random graphs,* Master's thesis, University of Waterloo, Canada, 1992.

[162] M. Molloy and B.A. Reed, *A critical point for random graphs with a given degree sequence,* Random Structures and Algorithms , to appear.

[163] R. Motwani, *Average case analysis of algorithms for matchings and related problems,* to appear.

[164] L.Perkovic and B.A.Reed, *Edge colouring random graphs in polynomial expected time*, to appear.

[165] A.D. Petford and D.J.A. Welsh, *A randomised 3-colouring algorithm,* Discrete Mathematics **74** (1989) 253-261.

[166] B. Pittel, *On the probable behavior of some algorithms for finding the stability number of a graph*, Mathematical Proceedings of the Cambridge Philosophical Society 92, 511-526.

[167] B. Pittel, *The average number of stable marriages*, SIAM Journal on Discrete Mathematics 2 (1989) 530-549.

[168] B. Pittel, *On likely solutions of a stable marriage problem*, Annals of Applied Probability 2 (1992) 358-401.

[169] B. Pittel, *The "stable roommates" problem with random preferences*, Annals Probability 21 (1993) 1441-1477.

[170] B. Pittel, *On a random instance of the "stable roommates" problem. Likely behaviour of the proposal algorithm*, Combinatorics, Probability and Computing 2 (1993) 53-92.

[171] B.Pittel and R.S.Weishar, *Online coloring of sparse random graphs and random trees*, to appear.

[172] B. Pittel, J. Spencer, N. Wormald, *Sudden emergence of a giant k-core in a random graph* Journal of Combinatorial Theory B, to appear.

[173] L. Pósa, *Hamiltonian circuits in random graphs*, Discrete Mathematics 14 (1976) 359-364.

[174] H.J. Prömel and A. Steger, *Coloring clique-free graphs in linear expected time,* Random Structures and Algorithms **3** (1992) 375-402.

[175] L.E.Rasmussen, *Approximating the permanent*, to appear.

[176] N. Robertson and P.D. Seymour, *Graph minors-XIII: The disjoint paths problem*, to appear in Journal of Combinatorial Theory B.

[177] R.W. Robinson and N.C. Wormald, *Almost all cubic graphs are Hamiltonian*, Random Structures and Algorithms 3 (1992) 117-126.

[178] R.W. Robinson and N.C. Wormald, *Almost all regular graphs are Hamiltonian*, Random Structures and Algorithms 5, (1994) 363-374.

[179] J.P. Schmidt, *Probabilistic analysis of strong hypergraph coloring algorithms and the strong chromatic number,* Discrete Mathematics 66 (1987) 259-277.

[180] J. Schmidt-Pruzan, E. Shamir and E. Upfal, *Random hypergraph coloring algorithms and the weak chromatic number,* J. Graph Theory 8 (1985) 347-362.

[181] C.P. Schnorr, *An algorithm for transitive closure with linear expected time,* SIAM J. Computing 7 (1978) 127-133.

[182] E. Shamir, *How many random edges make a graph Hamiltonian?*, Combinatorica 3 (1983) 123-132.

[183] E. Shamir and J. Spencer, *Sharp concentration of the chromatic number on random graphs* $G_{n,p}$*,* Combinatorica **7** (1987) 121-129.

[184] E. Shamir and E. Upfal, *N-processors graphs distributively achieve perfect matchings in* $O(log^2(N))$ *beats*, -Proceedings of the 1st ACM-PODC conference (1982) 238-241.

[185] E. Shamir and E. Upfal, *Sequential and distributed graph colouring algorithms with performance analysis in random graph spaces*, Journal of Algorithms 5 (1984) 488-501.

[186] E. Shamir and E. Upfal, *A fast construction of disjoint paths in networks*, Annals of Discrete Mathematics 24 (1985) 141-154.

[187] S.Shapley and H.Scarfe, *On cores and indivisibility*, Journal of mathematical Economics 1 (1974) 23-37.

[188] P.M. Spira, *A new algorithm for finding all shortest paths in a graph of positive arcs in average time* $O(n \log n)$, SIAM Journal on Computing 2 (1973) 28-32.

[189] J.M.Steele, *Probabilistic and worst-case analyses of classical problems of combinatorial optimization in Euclidean space*, Mathematics of Operations Research 15 (1990) 749-770.

[190] J.M.Steele, *On Frieze's* ζ *limit for lengths of minimum spanning trees*, Discrete Applied Mathematics 18 (1987) 99-103.

[191] C.R.Subramanian, *Minimum colouring random and semi-random graphs with polynomial expected time*, Proceedings of the 36th Annual IEEE Symposium on the Foundations of Computer Science (1995) 463-472.

[192] S.Suen, *Ph.D. Dissertation*, University of Bristol, 1985.

[193] A.Thomason, *A simple linear expected time algorithm for Hamilton cycles*, Discrete Mathematics 75 (1989) 373-379.

[194] A.Thomason, *Random graphs, strongly regular graphs and pseudo-random graphs*, in Surveys in Combinatorics 1987, (C.Whitehead Ed.), Cambridge University Press 1987.

[195] E.A. Timofeev, *On finding the expected length of a random minimal tree*, Theory of Probability and its Applications 32 (1987) 361-365.

[196] G. Tinhofer, *A probabilistic analysis of some greedy cardinality matching algorithms*, Annals of Operations Research 1 (1984) 239-254.

[197] N. Tomizawa, *On some techniques useful for solution of transportation problems*, Networks 1 (1972) 173-194.

[198] J.S. Turner, *Almost all k-colourable graphs are easy to color,* J. Algorithms **9** (1988) 63-82.

[199] W.T. Tutte, *The factorization of linear graphs*, Journal of the London Mathematical Society 22 (1947) 107-111.

[200] L.G. Valiant, *The complexity of computing the permanent*, Theoretical Computer Science 8 (1979) 189-201.

[201] L.G. Valiant and G.J. Brebner, *Universal schemes for parallel computation*, Proceedings of the 13th Annual ACM Symposium on Theory of Computing (1981) 263-277.

[202] D.W. Walkup, *Matchings in random regular bipartite graphs*, Discrete Mathematics 31 (1980) 59-64.

[203] D.W. Walkup, *On the expected value of a random asignment problem*, SIAM J. Computing 8 (1979) 440-442.

[204] H. Wilf, *Backtrack: an $O(1)$ expected time algorithm for the graph coloring problem,* Information Processing Letters 18 (1984) 119-121.

[205] N.C.Wormald, *Differential equations for random processes and random graphs*, to appear in Random Structures and Algorithms .

[206] J. Zerovnik, *A randomised heuristical algorithm for estimating the chromatic number of a graph,* Information Processing Letters 33 (1989) 213-219.

[207] J. Zerovnik, *A randomised algorithm for k-colorability,* Discrete Mathematics 131 (1994) 379-393.